Н. Н. Писарук

Исследование операций

Минск
2012

Писарук, Н. Н.

Исследование операций / Н. Н. Писарук. — Минск : БГУ, 2012. — 281 с.

В учебном пособии изучаются модели и методы из таких разделов исследования операций как нелинейная оптимизация с ограничениями, квадратичное, линейное, динамическое, целочисленное и стохастическое программирование, сетевая оптимизация, теория массового обслуживания. Вспомогательные сведения из других разделов математики приведены в приложениях.

Для студентов экономических, математических и инженерных специальностей университетов.

Оглавление

Предмет исследования операций

Исследование операций[1] (сокращенно *ИСО*) изучает применения количественных методов для управления сложными системами людей, машин, материалов, денег и информации. Методология исследования операций позволяет понять сущность управленческих проблем и разработать модели для оценки последствий принимаемых решений.

Исследование операций как самостоятельная научная дисциплина возникла в годы второй мировой войны, когда для решения сложных проблем логистики и проектирования систем вооружений создавались команды практиков, в которые входили специалисты из самых различных дисциплин: математики, инженеры, экономисты, психологи и т. д. Эти команды анализировали и формулировали проблему в количественных терминах, чтобы найти ее оптимальное решение. Сегодня методы исследование операций широко используются в *операционном менеджменте*[2] и других бизнес ориентированных дисциплинах.

0.1. Общая задача исследования операций

Мы можем записать общую задачу исследования операций следующим образом:

$$\begin{aligned} &f(x,y,z) \to \mathrm{ext} \\ &x \in X,\ y \in Y,\ z \in Z, \end{aligned} \tag{1}$$

где

[1] Интересно отметить, что в США в качестве синонима термина «operation research (исследование операций)» часто используется термин «management science».

[2] Операционный менеджмент можно определить как управление ресурсами (трудовыми, сырьевыми, финансовыми ресурсами, оборудованием и т. д.) при производстве продуктов или предоставлении услуг.

- x — вектор контролируемых факторов,
- y — вектор случайных факторов,
- z — вектор неопределенных факторов.

Значение контролируемых факторов выбирается теми, кто принимает решение (оперирующей стороной). Случайные и неопределенные факторы — это неконтролируемые факторы для оперируюшей стороны. Разница между случайными и неопределенными факторами состоит в следующем. Компоненты вектора y — это случайные величины с известным законом распределения. Например, y_5 есть нормальная случайная величина с матожиданием $m \in [m_1, m_2]$ и стандартным отклонением $\sigma \in [\sigma_1, \sigma_2]$. В противоположность, оперирующей стороне известны только области значений неопределенных факторов. Например, переменная z_3 принимает значения из отрезка $[1, 7]$.

Важными разделами исследования операций являются:

- математическое программирование ($X \neq \emptyset$, $Y = Z = \emptyset$);
- стохастическое программирование ($X \neq \emptyset$, $Y \neq \emptyset$, $Z = \emptyset$);
- теория игр ($X \neq \emptyset$, $Z \neq \emptyset$).

Пример 0.1 (минимизация упущенной выгоды). *В летний период спрос на гостиничные номера в курортном городе существенно превышает предложение. Владелец небольшой гостиницы хочет минимизировать недололученную прибыль из-за того, что очень часто ряд номеров в его гостинице пустует, поскольку приезжают не все клиенты, забронировавшие номера. Известна следующая статистика: все клиенты приезжают с вероятностью* 0.4*, ровно один клиент не приежает с вероятностью* 0.3*, ровно два клиента не приежают с вероятностью* 0.2*, ровно три клиента не приежают с вероятностью* 0.1*.*

Хозяин решил принимать заказов на резервирование номеров больше, чем имеется номеров в гостинице. Стоимость одного номера \$70*. В случае, если явятся больше клиентов, чем имеется мест, каждого лишнего клиента можно поселить в большой и дорогой гостинице со стоимость номера* \$120*; разницу в* $120 - 70 = 50$ *долларов хозяин должен клиенту компенсировать.*

Сколько лишних заявок нужно принимать, чтобы минимизировать ожидаемую недололученную прибыль?

Решение. Здесь владелец гостиницы должен принать решение $x \in X \stackrel{\text{def}}{=} \{0, 1, 2, 3\}$, где значение x — это количество принятых лишних заявок. Значение случайного фактора $y \in Y \overset{\text{def}}{=} \{0,1,2,3\}$ — это количество не явившихся клиентов. Определим *функцию потерь* $f : X \times Y \to \mathbb{R}$ по правилу:

$$f(x,y) \overset{\text{def}}{=} \begin{cases} 70(y-x), & \text{если } x < y, \\ 50(x-y), & \text{если } x \geq y. \end{cases}$$

При фиксированном значении x функция $f(x,y)$ есть дискретная случайная величина, которая принимает значение $f(x,0)$ с вероятностью 0.4, значение $f(x,1)$ с вероятностью 0.3, значение $f(x,2)$ с вероятностью 0.2, значение $f(x,3)$ с вероятностью 0.1. Вычислим математические ожидания $g(x) \overset{\text{def}}{=} E_y(f(x,y))$ для всех значений $x \in X$:

$$\begin{aligned} g(0) &= 0.4f(0,0) + 0.3f(0,1) + 0.2f(0,2) + 0.1f(0,3) = \\ &= 0.4 \cdot 0 + 0.3 \cdot 70 + 0.2 \cdot 140 + 0.1 \cdot 210 = 70, \\ g(1) &= 0.4f(1,0) + 0.3f(1,1) + 0.2f(1,2) + 0.1f(1,3) = \\ &= 0.4 \cdot 50 + 0.3 \cdot 0 + 0.2 \cdot 70 + 0.1 \cdot 140 = 48, \\ g(2) &= 0.4f(2,0) + 0.3f(2,1) + 0.2f(2,2) + 0.1f(2,3) = \\ &= 0.4 \cdot 100 + 0.3 \cdot 50 + 0.2 \cdot 0 + 0.1 \cdot 70 = 62, \\ g(3) &= 0.4f(3,0) + 0.3f(3,1) + 0.2f(3,2) + 0.1f(3,3) = \\ &= 0.4 \cdot 150 + 0.3 \cdot 100 + 0.2 \cdot 50 + 0.1 \cdot 0 = 100. \end{aligned}$$

Функция ожидаемых потерь $g(x)$ принимает минимальное значение 48 при $x = 1$. Значит, владелец гостиницы должен принимать всего один лишний заказ. □

Пример 0.2. *Издатель при продаже некоторой книги получает прибыль a и теряет b с каждой непроданной книги. Спрос на книги подобного жанра есть случайная величина с плотностью h. Максимально возможный спрос на книгу равен u. Сколько книг нужно издать, чтобы максимизировать ожидаемую прибыль?*

Решение. Если издатель выпускает $x \in X \overset{\text{def}}{=} [0,u]$ книг при спросе $y = Y \overset{\text{def}}{=} [0,u]$[3], то его прибыль равна

$$f(x,y) \overset{\text{def}}{=} \begin{cases} ay - b(x-y) = (a+b)y - bx, & \text{если } y < x, \\ ax, & \text{если } y \geq x. \end{cases}$$

[3] При достаточно больших тиражах мы можем не требовать, чтобы x и y принимали только целые значения.

Математическое ожидание величины прибыли равно

$$\begin{aligned}E_y(f(x,y)) &= \int_0^x ((a+b)z - bx)\, h(z)dz + \int_x^u axh(z)dz = \\ &= (a+b)\int_0^x zh(z)dz - bx\int_0^x h(z)dz + \\ &+ ax\int_0^u h(z)dz - ax\int_0^x h(z)dy = \\ &= (a+b)\int_0^x yh(z)dz - (a+b)x\int_0^x h(z)dz + ax.\end{aligned}$$

Максимизируя функцию $g(x) \stackrel{\text{def}}{=} E_y(f(x,y))$ по x, мы определим, сколько книг нужно выпустить. Запишем критерий оптимальности первого порядка:

$$g'(x) = (a+b)xh(x) - (a+b)xh(x) - (a+b)\int_0^x h(z)dz + a = 0.$$

Итак, мы найдем оптимальный выпуск x, решая уравнение

$$\int_0^x h(z)dz = \frac{a}{a+b}.$$

Для примера, если случайная величина y равномерно распределена на отрезке $[0,u]$, то $h(z) = 1/u$, $\int_0^x h(z)dz = x/u$ и $x = au/(a+b)$. □

Пример 0.3. *Человек взял велосипед напрокат в пункте A и выехал на прогулку. После всего трех минут езды его велосипед сломался, и человек решил катить его к ближайшему из трех пунктов проката A, B или C (см. рис. 1), где можно заменить сломанный велосипед на исправный. Человек знает, что он ехал со скоростью от 25 до 30 км/час, но он не помнит, в какую сторону он повернул на развилке (в сторону пункта B или пункта C).*

В каком направлении должен двигаться человек, чтобы пройденное им расстояние было минимальным в худшем из возможных случаев?

Решение. В данной ситуации человек может принять одно из двух решений $x \in X \stackrel{\text{def}}{=} \{1,2\}$, где $x = 1$ означает двигаться вперед, $x = 1$ — возвращаться назад. Здесь неопределенным фактором является местоположение человека, которое можно представить двумерным вектором

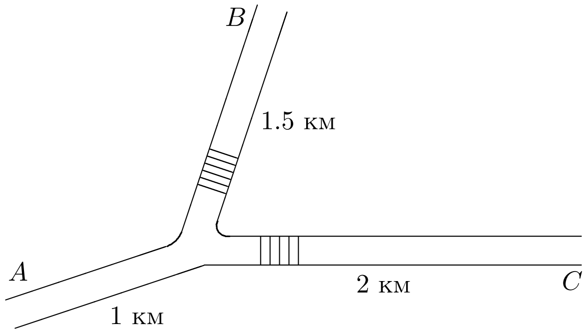


Рис. 1. Схема велосипедных маршрутов

$z = (z_1, z_2)^T$, где z_1 — это расстояние до пункта A, а z_2 принимает только два значения: 1 для обозначения того, что человек на развилке повернул в сторону пункта B, и 2 для обозначения того, что человек повернул в сторону пункта C.

Если человек ехал со скоростью 25 км/час, то он проехал $25 \cdot (3/60) = 1.25$ км, а если он ехал со скоростью 30 км/час, то он проехал $30 \cdot (3/60) = 1.5$ км. Поэтому $z_1 \in [1.25, 1.5]$ и $z \in Z \stackrel{\text{def}}{=} [1.25, 1.5] \times \{1, 2\}$.

Если человек, который находится в позиции $z \in Z$, принял решение $x \in X$, то ему придется пройти расстояние

$$f(x,z) \stackrel{\text{def}}{=} \begin{cases} 2.5 - z_1, & \text{если } x = 1 \text{ и } z_2 = 1, \\ 3 - z_1, & \text{если } x = 1 \text{ и } z_2 = 2, \\ z_1, & \text{если } x = 2. \end{cases} \tag{2}$$

Чтобы из двух своих решений, $x = 1$ или $x = 2$, найти оптимальное, человек должен решить следующую оптимизационную задачу:

$$\min_{x \in X} \max_{z \in Z} f(x,z) = \min_{x \in X} g(x),$$

где

$$g(x) \stackrel{\text{def}}{=} \max_{z \in Z} f(x,z).$$

С учетом (2) вычисляем

$$g(1) = \max\left\{\max_{1.25\le z_1\le 1.5} 2.5 - z_1, \max_{1.25\le z_1\le 1.5} 3 - z_1\right\} =$$
$$= \max\{1.25, 1.75\} = 1.75,$$
$$g(2) = \max_{1.25\le z_1\le 1.5} z_1 = 1.5.$$

Теперь ясно, что человеку лучше принять решение $x = 2$, т. е. он должен повернуть обратно и двигаться в пункт A. □

0.2. Упражнения

0.1. Фирма, производящая прохладительные напитки, продает 1 млн. литров в год, имея прибыль 0.25 доллара за литр. Владельцы крупной торговой сети предложили фирме производить в год 250 тыс. литров нового напитка, который будет продаваться в магазинах этой сети. Торговая сеть гарантируе фирме прибыль 0.15 доллара за литр. Если фирма откажется от предложения производить новый напиток, то его может принять одна из фирм-конкурентов. Выпуск нового напитка (нашей фирмой или одним из ее конкурентов) приведет к снижению потребления старого (производимого ныне) продукта.

В случае согласия производить новый продукт, спрос на старый продукт сократиться на 10 % с вероятностью 0.7, на 20 % с вероятностью 0.2, на 30 % с вероятностью 0.1

При отказе от предложения вероятность того, что какой-либо из конкурентов согласиться производить новый напиток равна 0.5. Если какой-либо из конкурентов начнет производить новый напиток, наша фирма может

а) ничего не предпринимать для сохранения существующего спроса на старый продукт, и тогда спрос сократится на 10 % с вероятностью 0.7, на 20 % с вероятностью 0.2, на 30 % с вероятностью 0.1;

б) увеличить на 25 тыс. долларов расходы на рекламу старого напитка, и тогда спрос сохранится на прежнем уровне с вероятностью 0.3, сократится на 5 % с вероятностью 0.4, на 10 % с вероятностью 0.3;

в) снизить цену на старый продукт, сократив прибыль до 0.2 долларов за литр; в таком случае с вероятностью 0.3 фирма-конкурент также снизит цену на новый продукт. Если фирмы снизят цены, то объем сбыта нашей фирмы сократится на 5 % с вероятностью 0.5, на 10 % с вероятностью 0.2, на 15 % с вероятностью 0.3. Если только наша фирма снизит цены, то объем сбыта нашей фирмы не изменится с вероятностью 0.3, сократится на 5 % с вероятностью 0.5, на 10 % с вероятностью 0.2.

Как должна действовать наша фирма, чтобы максимизировать свою прибыль?

Глава 1
Нелинейная оптимизация с ограничениями

1.1. Необходимые условия оптимальности

Будем рассматривать задачу

$$\begin{aligned} f(x) &\to \min, \\ g_i(x) &\le 0, \quad i \in I = \{1, \dots, m\}, \\ x &\in \mathbb{R}^n, \end{aligned} \tag{1.1}$$

где функции f и g_i $(i \in I)$ непрерывны и дифференцируемы. Обозначим через X множество решений задачи (1.1), т. е.

$$X = \{x \in \mathbb{R}^n : \ g_i(x) \le 0, \ i \in I\}.$$

1.1.1. Допустимые направления и выделение ограничений

Будем предполагать, что множество X непусто; однако допускаем, что оно может иметь пустую внутренность.

Точка $x^0 \in X$ есть *локальный оптимум* (*минимум*) задачи (1.1), если для некоторого числа $\epsilon > 0$ выполняется условие

$$f(x^0) \le f(x) \quad \forall\, x \in X, \ \|x - x^0\| \le \epsilon.$$

Если $x^0 \in X$ есть локальный оптимум задачи (1.1), то функция $f(x)$ не может убывать, когда x описывает дугу кривой (достаточно регулярной), выходящую из x^0 и содержащуюся в множестве решений X.

Такую дугу кривой будем называть допустимой и будем определять посредством непрерывно дифференцируемой функции $\varphi : \mathbb{R}_+ \to \mathbb{R}^n$ параметра $\theta \geq 0$:

$$\varphi(\theta) = [\varphi_1(\theta), \ldots, \varphi_n(\theta)],$$

которая удовлетворяет условиям

a) $\varphi(0) = x^0$;

b) $\varphi(\theta) \in X$ для достаточно малого $\theta > 0$.

Допустимым направлением в точке x^0 назовем вектор

$$y = \frac{d\varphi}{d\theta}(0) = \left[\frac{d\varphi_1}{d\theta}(0), \frac{d\varphi_2}{d\theta}(0), \ldots, \frac{d\varphi_n}{d\theta}(0)\right]^T,$$

касающийся некоторой дуги кривой $\varphi(\theta)$, допустимой в x^0.

В дальнейшем будем обозначать через $C_{\mathrm{ad}}(x^0)$ конус, образованный множеством допустимых направлений в точке x^0. Отыщем условие, необходимое для того, чтобы вектор $y \in \mathbb{R}^n$ принадлежал конусу $C_{\mathrm{ad}}(x^0)$.

Обозначим через $I(x^0)$ множество индексов *насыщенных* ограничений в x^0, т. е. ограничений, выполняющихся в x^0 в форме равенства:

$$I(x^0) = \{i \in I : \; g_i(x^0) = 0\}.$$

Касательный конус в точке x^0 определяется следующим образом:

$$T(x^0) \stackrel{\text{def}}{=} \{y \in \mathbb{R}^n : \; \nabla g_i^T(x^0) y \leq 0 \quad \forall\, i \in I(x^0)\}.$$

Лемма 1.1. *Справедливо включение $C_{\mathrm{ad}}(x^0) \subseteq T(x^0)$.*

Доказательство. Пусть $\varphi(\theta)$ есть дуга допустимой кривой в x^0, а $y = \frac{d\varphi}{d\theta}(0)$ — допустимое направление в x^0. Для $i \in I(x^0)$ при достаточно малом $\theta > 0$ должно выполняться неравенство $g_i(x^0) \leq 0$. Разлагая функцию $g_i(\varphi(\theta))$ в ряд Тейлора в окрестности точки $\theta = 0$, получим неравенство

$$g_i(x^0) + \theta \nabla g_i^T(x^0) \frac{d\varphi}{d\theta}(0) + \theta o(\theta) \leq 0,$$

где $o(\theta) \to 0$ при $\theta \to 0$. Значит, в силу равенства $g_i(x^0) = 0$, необходимо (но не достаточно), чтобы направление $y = \frac{d\varphi}{d\theta}(0)$ удовлетворяло условию

$$\nabla g_i^T(x^0) y \leq 0 \quad \forall\, i \in I(x^0).$$

Итак, из $y \in C_{\text{ad}}(x^0)$ следует, что $y \in T(x^0)$, и лемма доказана. $\square$

К сожалению, как показывает следующий пример обратное включение $T(x^0) \subseteq C_{\text{ad}}(x^0)$ в общем случае неверно.

Рассмотрим в $\mathbb{R}^2$ множество X, определяемое ограничениями

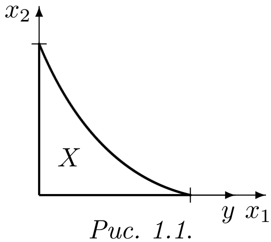


Рис. 1.1.

$$
\begin{aligned}
g_1(x) &= -x_1 \le 0,\\
g_2(x) &= -x_2 \le 0,\\
g_3(x) &= -(1-x_1)^3 + x_2 \le 0
\end{aligned}
$$

(см. рис 1.2). В точке $x^0 = (1,0)^T$ выполняются как равенства второе и третье ограничения. Значит, $I(x^0) = \{2,3\}$. Поскольку

$$\nabla g_2(x^0) = \begin{bmatrix} 0 \\ -1 \end{bmatrix}, \quad \nabla g_3(x^0) = \begin{bmatrix} 0 \\ 1 \end{bmatrix},$$

то неравенства, определяющие $T(x^0)$, будут следующими

$$-y_2 \le 0, \quad y_2 \le 0.$$

Вектор $y = (1,0)^T$ удовлетворяет этим неравенствам, однако для любой непрерывно дифференцируемой функции $\varphi : \mathbb{R}_+ \to \mathbb{R}^n$, что $\varphi(0) = x^0$ и $y = \frac{d\varphi}{d\theta}(0)$, для достаточно малых $\theta > 0$ точка

$$\varphi(\theta) = \varphi(0) + \theta \frac{d\varphi}{d\theta}(0) + \theta o(\theta)\mathrm{e} = x^0 + \theta y + \theta\, o(\theta)\mathrm{e} = \begin{bmatrix} 1 + \theta(1 + o(\theta)) \\ \theta\, o(\theta) \end{bmatrix}$$

не принадлежит X.

Заметим, что векторы $\nabla g_2(x^0)$ и $\nabla g_3(x^0)$ линейно зависимы!

Говорят, что в точке $x^0 \in X$ выполняется *условие выделения ограничений*, если касательный конус в этой точке является замыканием конуса допустимых направлений:

$$\mathrm{cl}(C_{\text{ad}}(x^0)) = T(x^0). \tag{1.2}$$

Выполнение условия выделения ограничений в точке x^0 означает, что конус допустимых направлений в точке x^0 совпадает с множеством решений y системы неравенств

$$\nabla g_i^T(x^0) y \le 0 \quad \forall\, i \in I(x^0).$$

На практике проверка выполнения условия (1.2) может оказаться трудной задачей. Поэтому были получены несколько достаточных условий, при выполнении которых равенство (1.2) имеет место. Наиболее важные результаты сформулированы в следующей лемме.

Лемма 1.2. *Условие выделения ограничений выполняется в каждой точке множества X в следующих случаях:*

а) все функции g_i линейны;

б) все функции g_i выпуклы и множество X имеет непустую внутренность.

Условие выделения ограничений выполняется в точке $x^0 \in X$, если

в) градиенты $\nabla g_i(x^0)$ ограничений, которые в точке x^0 выполняются как равенства, линейно независимы.

Доказательство. Обоснование условия а) элементарно. Поэтому мы ограничимся обоснованием условий б) и в).

Рассмотрим конус *допустимых направлений* в точке $x^0 \in X$:

$$C_{\mathrm{fs}}(x^0) \overset{\mathrm{def}}{=} \{y \in \mathbb{R}^n : \left(\nabla g_i(x^0)\right)^T y < 0,\ i \in I(x^0)\}.$$

Понятно, что $C_{\mathrm{fs}}(x^0) \subset C_{\mathrm{ad}}(x^0)$ и $\mathrm{cl}(C_{\mathrm{fs}}(x^0)) \subset \mathrm{cl}(C_{\mathrm{ad}}(x^0))$. В силу леммы 1.1 для завершения доказательства достаточно показать, что при выполнении условий б) и в) справедливо равенство $\mathrm{cl}(C_{\mathrm{fs}}) = T(x^0)$.

Сначала докажем, что нужное равенство справедливо, если $C_{\mathrm{fs}}(x^0) \neq \emptyset$. Действительно, для $\bar{y} \in C_{\mathrm{fs}}(x^0)$ выполняются неравенства:

$$\left(\nabla g_i(x^0)\right)^T \bar{y} < 0, \quad i \in I(x^0). \tag{1.3}$$

Для $y \in C_{\mathrm{fs}}(x^0)$ выполняются неравенства:

$$\left(\nabla g_i(x^0)\right)^T \bar{y} \le 0, \quad i \in I(x^0).$$

Поэтому $y(\lambda) \overset{\mathrm{def}}{=} (1-\lambda)\bar{y} + \lambda y \in C_{\mathrm{fs}}(x^0)$ для любого $\lambda \in [0,1)$. Следовательно, для любой неотрицательной последовательности $\{\lambda_k\}_{k=1}^{\infty}$ сходящейся к 1 слева ($\lambda_k < 1$), последовательность $\{x(\lambda_k)\}_{k=1}^{\infty}$ направлений из $C_{\mathrm{fs}}(x^0)$ сходится к направлению y. Это означает, что $\mathrm{cl}(C_{\mathrm{fs}}) = T(x^0)$.

Предположим теперь, что выполняется условие б). Тогда существует $\bar{x} \in X$, что

$$g_i(\bar{x}) < 0, \quad i \in I.$$

В силу выпуклости функций g_i справедливо неравенство:

$$\left(\nabla g_i(x^0)\right)^T(\bar{x}-x^0) \le g_i(\bar{x}) - g_i(x^0) = g_i(\bar{x}) < 0, \quad i \in I(x^0).$$

Это значит, что $\bar{y} = \bar{x} - x^0 \in C_{\mathrm{fs}}(x^0)$ и, следовательно, $C_{\mathrm{fs}}(x^0) \neq \emptyset$.

Теперь предположим, что в точке x^0 выполняется условие в). Тогда векторное равенство

$$\sum_{i\in I(x^0)} \lambda_i \nabla g_i(x^0) = 0$$

несовместна относительно неизвестных λ_i. По лемме Фаркаша (лемма A.1) существует такой вектор $y \in \mathbb{R}^n$, что выполняется неравенство (1.3) и, значит, $C_{\mathrm{fs}}(x^0) \neq \emptyset$. $\square$

Замечание 1. Условие б) леммы 1.2 можно расширить на случай, когда функции g_i являются псевдовыпуклыми.

Замечание 2. Различные условия леммы 1.2 допускают возможность комбинирования. Например, условие выделения ограничений выполняются в любой точке $x \in X$, если часть функций g_i линейные ($i \in I_l$), а остальные функции выпуклые ($i \in I_c$) и существует точка $\bar{x} \in X$, что $g_i(\bar{x}) < 0$ для всех $i \in I_c$.

Условие выделения ограничений выполняются в точке $x : 0 \in X$, если часть функций g_i линейные ($i \in I_l$), а градиенты $\nabla g_i(x^0)$ линейно независимы для нелинейных ограничений $i \in I_n$. и существует точка $\bar{x} \in X$, что $g_i(\bar{x}) < 0$ для всех $i \in I_n$.

1.1.2. Необходимые условия Куна — Таккера

Теорема 1.1 (Куна — Таккера). *Предположим, что все функции f и g_i ($i = 1, \ldots, m$) непрерывно дифференцируемы и в точке $x^0 \in X$ выполняется условие выделения ограничений. Если x^0 есть точка локального минимума, то существуют такие числа $\lambda_i \ge 0$ ($i = 1, \ldots, m$), что*

$$\nabla f(x^0) + \sum_{i=1}^{m} \lambda_i \nabla g_i(x^0) = 0, \tag{1.4}$$

$$\lambda_i g_i(x^0) = 0, \quad i = 1, \ldots, m. \tag{1.5}$$

(*Числа λ_i называются множителями Куна-Таккера.*)

Доказательство. Точка x^0 не является локальным минимумом задачи (1.1), если существует такое допустимое направление, вдоль которого целевая функция убывает. Алгебраически это условие эквивалентно тому, что следующая система неравенств

$$\left(\nabla f(x^0)\right)^T y \leq -1, \tag{1.6}$$

$$\left(\nabla g_i(x^0)\right)^T y \leq 0, \quad i \in I(x^0). \tag{1.7}$$

имеет решение. Поэтому несовместность системы (1.6)–(1.7) — это необходимое условие того, что точка x^0 является локальным минимумом задачи (1.1).

По лемме Фаркаша (лемма A.1), система (1.6)–(1.7) несовместна тогда и только тогда, когда существуют неотрицательные числа λ_0 и λ_i ($i \in I(x^0)$), такие, что

$$\lambda_0 \nabla f(x^0) + \sum_{i=1}^{m} \lambda_i \nabla g_i(x^0) = 0,$$

$$-1 \cdot \lambda_0 + \sum_{i=1}^{m} 0 \cdot \lambda_i = -1.$$

Из последнего равенства следует, что $\lambda_0 = 1$.

Чтобы завершить доказательство, нужно установить справедливость условия (1.5). Для этого достаточно положить $\lambda_i = 0$ для всех $i \in I \setminus I(x^0)$. □

Замечание. Если какое-либо ограничение i в задаче (1.1) должно выполняться как равенство, то соответствующий ему множитель λ_i может быть любого знака, т. е. $\lambda_i \in \mathbb{R}$. Этот факт следует из того, что уравнение $g_i(x) = 0$ можно заменить двумя неравенствами $g_i(x) \leq 0$ и $-g_i(x) \leq 0$, которым ставятся в соответствие два неотрицательных множителя Куна — Таккера λ_i^+ и λ_i^-. Тогда в векторном равенстве (1.4) будут присутствовать два слагаемых $\lambda_i^+ \nabla g_i(x^0)$ и $-\lambda_i^- \nabla g_i(x^0)$, которые можно заменить их суммой $(\lambda_i^+ - \lambda_i^-)\nabla g_i(x^0)$. Вводя новый множитель $\lambda_i = \lambda_i^+ - \lambda_i^-$, мы вернемся к представлению (1.4), где ограничение $g_i(x) = 0$ представлено одним слагаемым $\lambda_i \nabla g_i(x^0)$, но со множителем λ_i произвольного знака.

Если в решаемой задаче присутствует ограничение вида $g_i(x) \geq 0$, то у нас имеется две альтернативы: представить это ограничение в «стандартном» виде $-g_i(x) \leq 0$, или при записи условий Куна — Таккера потребовать, чтобы множитель λ_i был неположителен ($\lambda_i \leq 0$).

Точка $x^0 \in X$, которая удовлетворяет системе (1.4) и (1.5) называется *стационарной точкой*. Другими словами, теорема 1.1 утверждает, что при выполнении условия выделения ограничений локальные минимумы следует искать среди стационарных точек.

Если все функции f и g_i $(i = 1, \dots, m)$ выпуклы, то, как мы увидим позже, тогда все стационарные точки являются глобальными минимумами.

1.1.3. Геометрическая и физическая интерпретация условий Куна — Таккера

Геометрически условия Куна — Таккера проиллюстрированы на рис. 1.2, где в точке x^0 локального минимума обращаются в равенство два неравенства $g_1(x) \le 0$ и $g_3(x) \le 0$; поэтому $I(x^0) = \{1, 3\}$. Вектор $-\nabla f(x^0)$ составляет тупой угол с любым вектором y из конуса допустимых направлений $C_{\mathrm{ad}}(x^0)$, который совпадает с касательным конусом $T(x^0)$. Это геометрическое условие алгебраически выражается так: вектор $-\nabla f(x^0)$ выражается в виде линейной комбинации векторов $\nabla g_i(x^0)$ $(i \in I(x^0))$ с положительными коэффициентами λ_i.

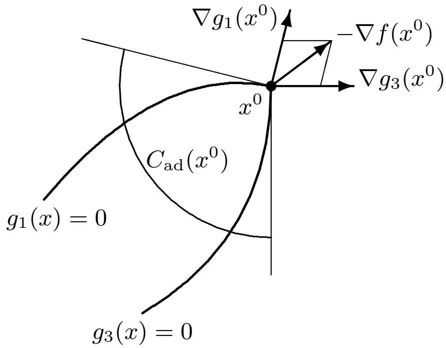


Рис. 1.2. Иллюстрация условий Куна — Таккера

Условия Куна — Таккера допускают также следующую физическую интерпретацию. *Материальная точка* движется внутри множества X под действием переменной силы, вектор которой в точке x равен $-\nabla f(x)$. Грани (границы) множества X являются абсолютно упругими и, когда материальная точка досгагает грани $g_i(x) = 0$ в точке x^0, на материальную точку дествует сила реакции $\lambda_i \nabla g_1(x^0)$, где множитель $\lambda_i \ge 0$ выбирается из условия, что сила $\lambda_i \nabla g_i(x^0)$ должна уравновешивать силу, с которой материальная точка давит на данную грань. Нужно найти *точку покоя* x^0, в которой движение материальной точки прекратиться. В такой интерпретации условия Куна — Таккера выражают тот факт, что в точке покоя силы реакции $\lambda_i \nabla g_i(x^0)$ граней уравновешивают силу

$-\nabla f(x^0)$, действующую на материальную точку.

1.1.4. Числовой пример

Записывая и решая системы уравнений и неравенств, выражающих условия Куна — Таккера, мы можем решать небольшие примеры оптимизационных задач. При этом следует заметить, что в компьютерных программах, способных решать задачи реалистичных для практики размеров, реализованы совершенно иные (численные) методы решения гладких оптимизационных задач с ограничениями, а теорема Куна — Таккера — это важный теоретический результат, который применяется при доказательтве многих теорем.

Пример 1.1. *Решить задачу*

$$\begin{aligned} f(x) &= x_1^2 + x_2^2 + x_3^2 \to \min, \\ g_1(x) &= 2x_1 - x_2 + x_3 - 5 \le 0, \\ g_2(x) &= x_1 + x_2 + x_3 - 3 = 0. \end{aligned}$$

Учитывая, что

$$\nabla f(x) = \begin{bmatrix} 2x_1 \\ 2x_2 \\ 2x_3 \end{bmatrix}, \quad \nabla g_1(x) = \begin{bmatrix} 2 \\ -1 \\ 1 \end{bmatrix}, \quad \nabla g_2(x) = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix},$$

запишем условия Куна — Таккера:

$$\begin{aligned} \begin{bmatrix} 2x_1 \\ 2x_2 \\ 2x_3 \end{bmatrix} + \lambda_1 \begin{bmatrix} 2 \\ -1 \\ 1 \end{bmatrix} + \lambda_2 \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} &= 0, \\ 2x_1 - x_2 + x_3 - 5 &\le 0, \\ x_1 + x_2 + x_3 - 3 &= 0, \\ \lambda_1(2x_1 - x_2 + x_3 - 5) &= 0, \\ \lambda_2(x_1 + x_2 + x_3 - 3) &= 0, \\ \lambda_1 &\ge 0. \end{aligned}$$

или

$$
\begin{aligned}
2x_1 + 2\lambda_1 + \lambda_2 &= 0,\\
2x_2 - \lambda_1 + \lambda_2 &= 0,\\
2x_3 + \lambda_1 + \lambda_2 &= 0,\\
2x_1 - x_2 + x_3 - 5 &\le 0,\\
\lambda_1(2x_1 - x_2 + x_3 - 5) &= 0,\\
x_1 + x_2 + x_3 &= 3,\\
\lambda_1 &\ge 0.
\end{aligned}
$$

Рассмотрим два случая.

$\lambda_1 = 0$. Тогда из первых трех уравнений получаем, что $x_1 = -\frac{\lambda_2}{2}$, $x_2 = -\frac{\lambda_2}{2}$ и $x_3 = -\frac{\lambda_2}{2}$. Подставляя эти значения в поледнее уравнение, найдем λ_2:

$$x_1 + x_2 + x_3 = -\frac{3}{2}\lambda_2 = 3 \quad \Rightarrow \quad \lambda_2 = -2.$$

Откуда $x^1 = (1,1,1)^T$ — стационарная точка. Причем, поскольку $f(x)$ — выпуклая функция, то x^1 точка глобального минимума[4].

$\lambda_1 > 0$. Теперь в силу условия дополняющей нежесткости

$$2x_1 - x_2 + x_3 = 5.$$

Их первых трех уравнений найдем

$$
\begin{aligned}
x_1 &= -\frac{1}{2}(2\lambda_1 + \lambda_2),\\
x_2 &= -\frac{1}{2}(-\lambda_1 + \lambda_2),\\
x_3 &= -\frac{1}{2}(\lambda_1 + \lambda_2).
\end{aligned}
$$

Подставляя эти значения в уравнения

$$
\begin{aligned}
2x_1 - x_2 + x_3 &= 5,\\
x_1 + x_2 + x_3 &= 3,
\end{aligned}
$$

[4] Этот факт будет установлен позже в Теореме 1.5.

получим

$$-2\lambda_1 - \lambda_2 - \frac{1}{2}\lambda_1 + \frac{1}{2}\lambda_2 - \frac{1}{2}\lambda_1 - \frac{1}{2}\lambda_2 = 5,$$
$$-\lambda_1 - \frac{1}{2}\lambda_2 + \frac{1}{2}\lambda_1 - \frac{1}{2}\lambda_2 - \frac{1}{2}\lambda_1 - \frac{1}{2}\lambda_2 = 3,$$

или

$$-3\lambda_1 - \lambda_2 = 5,$$
$$-\lambda_1 - \frac{3}{2}\lambda_2 = 3.$$

Умножив первое уравнение на $-\frac{3}{2}$ и сложив со вторым, получим

$$\left(\frac{9}{2} - 1\right)\lambda_1 + \left(\frac{3}{2} - \frac{3}{2}\right)\lambda_2 = -\frac{3}{2}5 + 3, \quad \text{или} \quad \frac{7}{2}\lambda_1 = -\frac{9}{2}.$$

Отсюда $\lambda_1 = -\frac{9}{7}$, что противоречит требованию неотрицательности λ_1.

Следовательно, $x^1 = (1, 1, 1)$ — единственная точка глобального минимума. □

1.1.5. Экономическая интерпретация множителей Куна — Таккера

Фирма использует n производственных процесса для производства n продуктов. Процес j $(j = 1, \ldots, n)$ описывается производственной функцией f_j:

$$x_j = f_j(x_1^j, \ldots, x_m^j),$$

где переменная x_j обозначает количество единиц продукта j, производимого j-м процессом, а переменная x_i^j обозначает количество единиц ресурса i $(i = 1, \ldots, m)$, используемого в j-м процессе. В наличии имеется a_i единиц ресурса i, $i = 1, \ldots, m$. Задан вектор цен $p = (p_1, \ldots, p_n)^T$ выпускаемых продуктов. Нужно найти производственный план $x^* = (x_1^*, \ldots, x_n^*)^T$, стоимость которого $p^T x^*$ максимальна.

Данная задача формулируется следующим образом:

$$p^T x \to \max, \tag{1.8a}$$

$$\lambda_j : \quad x_j - f_j(x_1^j, \ldots, x_m^j) = 0, \quad j = 1, \ldots, n, \tag{1.8b}$$

$$\mu_i: \quad \sum_{j=1}^{n} x_i^j - a_i \leq 0, \quad i = 1, \ldots, m, \tag{1.8c}$$

$$\nu_j: \quad x_j \geq 0, \quad j = 1, \ldots, n, \tag{1.8d}$$

$$\rho_i^j: \quad x_i^j \geq 0, \quad i = 1, \ldots, m, \ j = 1, \ldots, n. \tag{1.8e}$$

Здесь в самом левом столбце записаны множители Куна — Таккера для соответствующих ограничений.

Условия Куна — Таккера для задачи (1.8) записываются следующим образом:

$$-p_j + \lambda_j + \nu_j = 0, \quad j = 1, \ldots, n, \tag{1.9a}$$

$$\mu_i - \lambda_j \frac{\partial f_j}{\partial x_i^j}(x_1^j, \ldots, x_m^j) + \rho_i^j = 0, \quad i = 1, \ldots, m, \ j = 1, \ldots, n, \tag{1.9b}$$

$$x_j - f_j(x_1^j, \ldots, x_m^j) = 0, \quad j = 1, \ldots, n, \tag{1.9c}$$

$$\sum_{j=1}^{n} x_i^j \leq a_i\,, \quad i = 1, \ldots, m, \tag{1.9d}$$

$$\mu_i \left(\sum_{j=1}^{n} x_i^j - a_i \right) = 0, \quad i = 1, \ldots, m, \tag{1.9e}$$

$$\nu_j x_j = 0, \quad j = 1, \ldots, n, \tag{1.9f}$$

$$\rho_i^j x_i^j = 0, \quad i = 1, \ldots, m, \ j = 1, \ldots, n, \tag{1.9g}$$

$$\nu_j \leq 0, \quad j = 1, \ldots, n, \tag{1.9h}$$

$$\rho_i^j \leq 0, \quad i = 1, \ldots, m, \ j = 1, \ldots, n. \tag{1.9i}$$

Если продукт j производится ($x_j > 0$), то из условия дополняющей нежесткости (1.9f) имеем, что $\nu_j = 0$, и тогда из (1.9a) следует, что $\lambda_j = p_j$, т. е.

множители, соответствующие технологическим процессам производимых продуктов, равны ценам этих продуктов.

Если ресурс i используется в j-м процессе ($x_i^j > 0$), то из (1.9g) вытекает, что $\rho_i^j = 0$, и тогда для проиводимого продукта j ($x_j > 0$) из (1.9b) имеем:

$$\mu_i = p_j \frac{\partial f_j}{\partial x_i^j}(x_1^j, \ldots, x_m^j).$$

Если ресурс i не используется полностью $\left(\sum_{j=1}^{n} x_i^j < a_i\right)$, то из (1.9e) имеем, что $\mu_i = 0$. Но, если ресурс i используются в производственном процессе для какого-либо производимого продукта j, и поскольку $p_j > 0$ и $\frac{\partial f_j}{\partial x_i^j}(x_1^j, \ldots, x_m^j) > 0$, то и $\mu_i > 0$, т. е. такой ресурс i должен использоваться полностью.

Суммируя сказанное выше, мы формулируем свойства множителей ресурсных ограничений следующим образом:

> множитель ресурса, который не используется ни в одном технологическом процессе, производящем продукт, равен нулю; если ресурс i используется в технологическом процессе, производящем некоторый продукт j, то соответствующий этому ресурсу множитель μ_i равен стоимости предельного продукта j относительно ресурса i.

1.2. Достаточные условия оптимальности

В этом параграфе мы изучим достаточные условия оптимальности для задач следующего вида:

$$\begin{aligned} f(x) &\to \min, \\ g_i(x) &\le 0, \quad i \in I = \{1, \ldots, m\}, \\ x &\in S \subseteq \mathbb{R}^n. \end{aligned} \tag{1.10}$$

Заметим, что когда $S = \mathbb{R}^n$, то мы вновь приходим к задаче (1.1). Но теперь множество S может быть даже дискретным ($S \subseteq \mathbb{Z}^n$), и тогда мы будем иметь задачу целочисленного программирования.

1.2.1. Седловые точки и функции Лагранжа

Поставим в соответствие i-му ограничению ($i \in I$) неотрицательное действительное число, называемое *множителем Лагранжа*. Определим *функцию Лагранжа* по правилу:

$$L(x, \lambda) = f(x) + \sum_{i \in I} \lambda_i g_i(x).$$

Говорят, что точка $(\bar{x}, \bar{\lambda}) \in S \times \mathbb{R}_+^m$ есть *седловая точка* функции $L(x, \lambda)$, если

$$L(\bar{x}, \lambda) \le L(\bar{x}, \bar{\lambda}) \le L(x, \bar{\lambda}), \quad x \in S,\ \lambda \in \mathbb{R}_+^m. \tag{1.11}$$

Пример седловой точки для функции двух переменных приведен на рис. 1.3. Здесь $\bar{x}$ есть точка минимума функции $L(x, \bar{\lambda})$ по $x \in S$, а $\bar{\lambda}$ есть точка максимума функции $L(\bar{x}, \lambda)$ по $\lambda \in \mathbb{R}_+$. Можно представить, что трехмерная точка $(\bar{x}, \bar{\lambda}, L(\bar{x}, \bar{\lambda}))$ находится в центре поверхности седла для верховой езды на лошади.

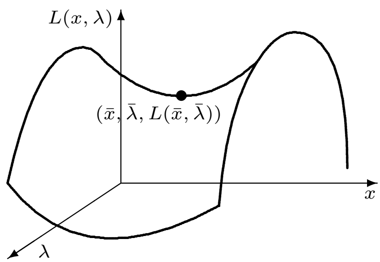


Рис. 1.3.

Теорема 1.2 (свойства седловых точек). *Точка* $(\bar{x}, \bar{\lambda}) \in S \times \mathbb{R}^m_+$ *является седловой для функции* $L(x, \lambda)$ *тогда и только тогда, когда*

$$L(\bar{x}, \bar{\lambda}) = \min_{x \in S} L(x, \bar{\lambda}), \tag{1.12a}$$

$$g_i(\bar{x}) \leq 0, \quad i \in I, \tag{1.12b}$$

$$\bar{\lambda}_i g_i(\bar{x}) = 0, \quad i \in I. \tag{1.12c}$$

Доказательство. Если $(\bar{x}, \bar{\lambda})$ — седловая точка, то равенство (1.12a) выполняется. С другой стороны, для любого $\lambda \in \mathbb{R}^m_+$ имеем неравенство

$$f(\bar{x}) + \sum_{i \in I} \bar{\lambda}_i g_i(\bar{x}) = L(\bar{x}, \bar{\lambda}) \geq L(\bar{x}, \lambda) = f(\bar{x}) + \sum_{i \in I} \lambda_i g_i(\bar{x}).$$

Откуда

$$\sum_{i \in I} (\lambda_i - \bar{\lambda}_i) g_i(\bar{x}) \leq 0, \quad \lambda \in \mathbb{R}^m_+. \tag{1.13}$$

Если условие (1.12b) не выполняется для некоторого индекса $i \in I$, то всегда можно выбрать достаточно большое $\lambda_i > 0$, чтобы не выполнялось неравенство (1.13). Поэтому все неравенства в (1.12b) должны выполняться.

Наконец при $\lambda = 0$ неравенство (1.13) превращается в неравенство

$$\sum_{i \in I} \bar{\lambda}_i g_i(\bar{x}) \geq 0.$$

Но поскольку $\bar{\lambda}_i \geq 0$ и $g_i(\bar{x}) \leq 0$ для всех $i \in I$, то

$$\sum_{i \in I} \bar{\lambda}_i g_i(\bar{x}) \leq 0.$$

Следовательно,

$$\sum_{i \in I} \bar{\lambda}_i g_i(\bar{x}) = 0$$

и поэтому

$$\bar{\lambda}_i g_i(\bar{x}) = 0 \quad \text{для всех } i \in I.$$

□

Теорема 1.3 (достаточное условие оптимальности). *Если пара* $(\bar{x}, \bar{\lambda}) \in S \times \mathbb{R}_+^m$ *есть седловая точка функции* $L(x, \lambda)$*, то* $\bar{x}$ *является глобальным оптимумом задачи* (1.10).

Доказательство. Для любого $x \in S$, удовлетворяющего условию $g_i(x) \leq 0$ $(i \in I)$, из условий (1.12a)–(1.12c) с учетом того, что $\lambda_i \geq 0$, имеем

$$f(\bar{x}) = f(\bar{x}) + \sum_{i \in I} \bar{\lambda}_i g_i(\bar{x}) \leq f(x) + \sum_{i \in I} \bar{\lambda}_i g_i(x) \leq f(x).$$

□

Теорема 1.3 — это весьма общий результат, который применим к любым задачам вида (1.10): выпуклым и невыпуклым, с дифференцируемыми и недифференцируемыми функциями f и g_i, непрерывными и дискретными множествами S. Неудивительно, что имеются задачи, для которых функция Лагранжа не имеет седловых точек.

Для примера, рассмотрим следующую задачу с одной переменной x:

$$\begin{aligned} &f(x) = -x^2 \to \min, \\ &g_1(x) = 2x - 1 \leq 0, \\ &S = \{x \in \mathbb{R} :\ 0 \leq x \leq 1\}. \end{aligned} \tag{1.14}$$

Поскольку функция Лагранжа

$$L(x, \lambda) = -x^2 + \lambda(2x - 1)$$

вогнута по x, то

$$\min_{x \in [0,1]} L(x, \lambda)$$

при любом фиксированном λ достигается либо в точке $x = 0$, либо в точке $x = 1$.

Поскольку единственный глобальный минимум в задаче (1.14) $x^* = 1/2$ не является точкой минимума функции $L(x, \lambda)$ ни при каком значении λ, то $L(x, \lambda)$ не имеет седловой точки.

1.2.2. Существование седловой точки для задач выпуклого программирования

Задача *выпуклого программирования* — это задача (1.10), когда все функции f и g_i $(i \in I)$ выпуклы, а множество S — также выпукло.

Теорема 1.4. *Пусть все функции f и g_i $(i \in I)$ выпуклы, множество S — также выпукло, и существует такая точка $x \in S$, что*

$$g_i(x) < 0, \quad i \in I. \tag{1.15}$$

Тогда если задача (1.10) имеет оптимальное решение $\bar{x}$, то существует такой вектор множителей $\bar{\lambda} \in \mathbb{R}^m_+$, что $(\bar{x}, \bar{\lambda})$ есть седловая точка функции Лагранжа $L(x, \lambda)$.

Доказательство. Пусть $\bar{x}$ — оптимальное решение задачи (1.10). Рассмотрим множества

$$A = \{(y_0, y) \in \mathbb{R} \times \mathbb{R}^m : \ \exists\, x \in S, \text{ что } y_0 \geq f(x),\ y_i \geq g_i(x), \quad i \in I\},$$
$$B = \{(y_0, y) \in \mathbb{R} \times \mathbb{R}^m : \ y_0 \leq f(\bar{x}),\ y_i \leq 0, \quad i \in I\}.$$

Оба эти множества выпуклы. То, что множество B выпукло, проверяется просто. Докажем, что множество A выпукло. Пусть $(y_0^1, y^1), (y_0^2, y^2) \in A$. Нам нужно доказать, что для любого $\delta \in [0, 1]$ точка

$$(\tilde{y}_0, \tilde{y}) \stackrel{\text{def}}{=} (1-\delta)(y_0^1, y^1) + \delta(y_0^2, y^2))$$

также принадлежит множеству A. Поскольку $(y_0^1, y^1), (y_0^2, y^2) \in A$, то существуют точки $x^1, x^2 \in S$, что

$$y_0^1 \geq f(x^1),\ y_i^1 \geq g_i(x^1), \quad i \in I,$$
$$y_0^2 \geq f(x^2),\ y_i^2 \geq g_i(x^2), \quad i \in I.$$

Откуда, с учетом выпуклости функций f и g_i, имеем неравенства

$$(1-\delta)y_0^1 + \delta y_0^2 \geq (1-\delta) f(x^1) + \delta f(x^2) \geq f((1-\delta)x^1 + \delta x^2),$$
$$(1-\delta)y_0^i + \delta y_i^2 \geq (1-\delta) g_(x^1) + \delta g_i(x^2) \geq g_i((1-\delta)x^1 + \delta x^2), \quad i \in I.$$

Из этих неравенств и определения множества A следует, что точка $(\tilde{y}_0, \tilde{y})$ также принадлежит множеству A.

Так как $f(\bar{x})$ есть оптимальное решение задачи (1.10), то пересечение $A \cap \operatorname{int} B$ пусто, и так как $\operatorname{int} B$ непусто, то существует гиперплоскость, разделяющая множества A и B, т. е. существует ненулевой вектор

$(u_0, u) \in \mathbb{R} \times \mathbb{R}^m$, что для любых $(y_0, y) \in A$ и $(z_0, z) \in B$ справедливо неравенство

$$u_0 y_0 + u^T y \geq u_0 z_0 + u^T z. \tag{1.16}$$

Покажем, что $u_0 > 0$ и все $u_i \geq 0$ $(i \in I)$. Действительно, если $u_i < 0$ для некоторого $i \in I$, то то мы можем уменьшить компоненту z_i так, чтобы неравенство (1.16) нарушилось. Поскольку $(f(\bar{x}), 0) \in A$ и $(f(x), g(x)) \in B$ для всех $x \in S$, то из (1.16) мы имеем

$$u_0 f(x) + u^T g(x) \geq u_0, \quad x \in S. \tag{1.17}$$

Если допустить, что $u_0 \leq 0$, то мы имели бы неравенства

$$u^T g(x) \geq 0, \quad x \in S,$$

что невозможно, так как существует такое $x \in S$, что $g_i(x) < 0$ для всех $i \in I$. Следовательно, $u_0 > 0$.

Положим $\bar{\lambda} = u/u_0$. Заметим, что $\bar{\lambda} \geq 0$. Из (1.17) имеем

$$f(x) + \bar{\lambda}^T g(x) \geq f(\bar{x}), \quad x \in S. \tag{1.18}$$

Полагая в (1.18) $x = \bar{x}$, получим неравенство $\bar{\lambda}^T g(\bar{x}) \geq 0$. Но одновременно справедливо и неравенство $\bar{\lambda}^T(\bar{x}) \leq 0$ (поскольку $\bar{\lambda} \geq 0$ и $g(x) \leq 0$). Следовательно, $\bar{\lambda}^T g(\bar{x}) = 0$.

Складывая равенство $0 = \bar{\lambda}^T g(\bar{x})$ со всеми неравенствами из (1.18), получим неравенства

$$f(x) + \bar{\lambda}^T g(x) \geq f(\bar{x}) + \bar{\lambda}^T g(\bar{x}), \quad x \in S,$$

или

$$L(x, \bar{\lambda}) \geq L(\bar{x}, \bar{\lambda}), \quad x \in S.$$

Теперь по теореме 1.2, $(\bar{x}, \bar{\lambda})$ есть седловая точка функции $L(x, \lambda)$. $\square$

1.2.3. Связь с условиями Куна — Таккера

Теорема 1.5. *Если в задаче (1.1) все функции f и g_i $(i \in I)$ выпуклы и непрерывно дифференцируемы, то для того чтобы точка $\bar{x}$ была глобальным минимумом, необходимо и достаточно, чтобы в точке $\bar{x}$ выполнялись условия Куна — Таккера.*

Доказательство. В силу теоремы 1.4 при $S = \mathbb{R}^n$ точка $\bar{x}$ есть глобальный минимум в том и только том случае, если существует $\bar{\lambda} \geq 0$, что $(\bar{x}, \bar{\lambda})$ есть седловая точка функции Лагранжа $L(x, \lambda)$. По теореме 1.2 должны выполняться следующие условия:

а) $\bar{x} \in \arg\min_{x \in \mathbb{R}^n} L(x, \bar{\lambda})$;

б) $g_i(\bar{x}) \leq 0$ для всех $i \in I$;

в) $\bar{\lambda}_i g_i(\bar{x}) = 0$ для всех $i \in I$.

Поскольку функции f и g_i выпуклы и дифференцируемы, то функция $L(x, \bar{\lambda})$ выпукла по x и, значит, условие а) равносильно векторному равенству

$$\nabla_x L(\bar{x}, \bar{\lambda}) = 0,$$

или

$$\nabla f(\bar{x}) + \sum_{i \in I} \bar{\lambda}_i \nabla g_i(\bar{x}) = 0,$$

которое в комбинации с б) и в) дает условия Куна — Таккера в точке $\bar{x}$.
□

Из доказательства теоремы 1.5 мы видим, что в дифференцируемом выпуклом случае множители Куна — Таккера отождествляются с множителями Лагранжа в седловой точке.

1.3. Геометрическое программирование

В этом разделе мы рассматриваем класс оптимизационных задач, которые не являются выпуклыми, но которые могут быть преобразованы в задачи выпуклого программирования заменой переменных и преобразованием целевой функции и функций в ограничениях.

1.3.1. Мономы и позиномы

Функция $f : \mathbb{R}^n_{++} \to \mathbb{R}$, определенная по правилу

$$f(x) \stackrel{\text{def}}{=} c x_1^{\alpha_1} x_2^{\alpha_2} \ldots x_n^{\alpha_n}, \tag{1.19}$$

называется *мономом*.

В дальнейшем мы будем предполагать, что коэффициенты α_i ($i = 1, \ldots, n$) могут быть любыми действительными числами, но коэффициент c должен быть положительным. Отметим, что такое допущение не совсем согласуется с определением монома в алгебре, где предполагается, что коэффициенты α_i должны быть положительными.

Сумма мономов

$$f(x) \stackrel{\text{def}}{=} \sum_{k=1}^{K} c_k x_1^{\alpha_{k1}} x_2^{\alpha_{k2}} \dots x_n^{\alpha_{kn}} \tag{1.20}$$

называется *позиномом*. Класс позиномов замкнут относительно сложения, умножения и деления на мономы.

1.3.2. Задача геометрического программирования

Оптимизационная задача вида

$$f(x) \to \min, \tag{1.21a}$$
$$g_i(x) \leq 1, \quad i = 1, \dots, p, \tag{1.21b}$$
$$h_i(x) = 1, \quad i = 1, \dots, q, \tag{1.21c}$$
$$x_j > 0, \quad j = 1, \dots, n, \tag{1.21d}$$

где $f, g_1, \dots, g_p$ есть позиномы, а $h_1, \dots, h_q$ — мономы, называется *задачей геометрического программирования*.

Можно сказать, что задача вида (1.21) есть стандартная форма для задачи геометрического программирования. В общем случае допускаются:

а) ограничения $v_i(x) \leq u_i(x)$, где $v_i(x)$ — позином, $u_i(x)$ — моном, которые можно записать в стандартной форме (1.21b) следующим образом: $g_i(x) \stackrel{\text{def}}{=} v_i(x)/u_i(x) \leq 1$;

б) ограничения $v_i(x) = u_i(x)$, где $v_i(x)$ и $u_i(x)$ — мономы, которые можно записать в стандартной форме (1.21c) следующим образом: $h_i(x) \stackrel{\text{def}}{=} v_i(x)/u_i(x) = 1$;

в) максимизация мономиальной целевой функции $\bar{f}(x)$, поскольку такая задача эквивалентна минимизации обратной целевой функции $f(x) \stackrel{\text{def}}{=} 1/\bar{f}(x)$, которая является мономом.

Для примера, задача

$$
\begin{aligned}
&x/y^2 \to \max,\\
&x^2 + 2y^2/\sqrt{z} \le y,\\
&x^3/y = z^2,\\
&2 \le y \le 5,\\
&x > 0,\ z > 0
\end{aligned}
$$

переписывается в стандартной форме (1.21) следующим образом:

$$
\begin{aligned}
&xy^{-2} \to \min,\\
&x^2y^{-1} + 2yz^{-1/2} \le 1,\\
&x^3y^{-1}z^{-2} = 1,\\
&2y^{-1} \le 1,\ (1/5)y \le 1,\\
&x > 0,\ z > 0.
\end{aligned}
$$

1.3.3. Сведение к задаче выпуклого программирования

Рассматриваем задачу (1.21). Пусть

$$
\begin{aligned}
f(x) &= \sum_{k=1}^{k_0} c_k^0 x_1^{\alpha_{k1}^0} x_2^{\alpha_{k2}^0} \dots x_n^{\alpha_{kn}^0},\\
g_i(x) &= \sum_{k=1}^{k_i} c_k^i x_1^{\alpha_{k1}^i} x_2^{\alpha_{k2}^i} \dots x_n^{\alpha_{kn}^i}, \quad i = 1, \dots, p,\\
h_i(x) &= \bar{c}_i x_1^{\bar{\alpha}_1^i} x_2^{\bar{\alpha}_2^i} \dots x_n^{\bar{\alpha}_n^i}, \quad i = 1, \dots, q,\\
b_{ik} &\stackrel{\text{def}}{=} \log c_k^i, \quad k = 1, \dots, k_i;\ i = 0, \dots, p,\\
a_{ik} &\stackrel{\text{def}}{=} (\log \alpha_{k1}^i, \dots, \log \alpha_{kn}^i)^T \quad k = 1, \dots, k_i;\ i = 0, \dots, p,\\
\bar{b}_i &\stackrel{\text{def}}{=} \log \bar{c}_i, \quad i = 1, \dots, q,\\
\bar{a}_i &\stackrel{\text{def}}{=} (\log \alpha_1^i, \dots, \log \alpha_n^i)^T \quad i = 1, \dots, q.
\end{aligned}
$$

Сделаем замену переменных $y_i = \log x_i$, тогда $x_i = \mathrm{e}^{y_i}$, $i = 1, \dots, n$. В новых переменных y_i задача (1.21) переписывается следующим образом:

$$\sum_{k=1}^{k_0} \mathrm{e}^{a_{0k}^T y + b_{0k}} \to \min,$$
$$\sum_{k=1}^{k_i} \mathrm{e}^{a_{ik}^T y + b_{ik}}, \quad i = 1, \dots, p,$$
$$\mathrm{e}^{\bar{a}_i^T y + \bar{b}_i}, \quad i = 1, \dots, q.$$

Логарифмируя целевую функцию и правые и левые части ограничений, в результате получим задачу

$$\tilde{f}(x) \stackrel{\text{def}}{=} \log \left(\sum_{k=1}^{k_0} \mathrm{e}^{a_{0k}^T y + b_{0k}} \right) \to \min, \tag{1.22a}$$

$$\tilde{g}_i(x) \stackrel{\text{def}}{=} \log \left(\sum_{k=1}^{k_i} \mathrm{e}^{a_{ik}^T y + b_{ik}} \right) \le 0, \quad i = 1, \dots, p, \tag{1.22b}$$

$$\tilde{h}_i(x) \stackrel{\text{def}}{=} \bar{a}_i^T y + \bar{b}_i = 0, \quad i = 1, \dots, q. \tag{1.22c}$$

Поскольку функция $\tilde{f}$ и все функции $\tilde{g}_i$ являются выпуклыми, то задача (1.22) является задачей выпуклого программирования.

1.4. Примеры задач нелинейного программирования

1.4.1. Неоклассическая задача потребления

Потребитель может потреблять n наборов благ (товары и услуги). Набор благ — это любой вектор $x \in \mathbb{R}^n_+$, где x_j есть количество блага j в наборе x. Потребитель описывается его функцией полезности $U : \mathbb{R}^n_+ \to \mathbb{R}$, которая

а) дважды непрерывно дифференцируема по всем n аргументам;

б) неубывающая: $\frac{\partial U(x)}{\partial x_j} \ge 0$ для всех $j = 1, \dots .n$;

в) вогнутая: в любой точке $x \in \mathbb{R}^n_{++}$ матрица вторых производных $\nabla^2 U(x)$ неположительно определена.

Заметим, что более строгий вариант свойства в), когда требуется строгая вогнутость функции U, подразумевает выполнение закона Госсена, который утверждает, что с ростом объема потребления любого блага j его предельная полезность убывает: $\frac{\partial^2 U}{\partial^2 x_j}(x) < 0$.

Неоклассическая задача потребления состоит в максимизации функции полезности на множестве потребления $\mathbb{R}^n_+$ при известных ценах $p \in \mathbb{R}^n_{++}$ и бюджете (доходе) потребителя I:

$$\max\{U(x) : \; p^T x \le I, \; x \ge 0\}. \tag{1.23}$$

Пусть x^0 есть решение задачи (1.23). По теореме Куна-Таккера существуют число $\lambda_0 \ge 0$ и вектор $\lambda \in \mathbb{R}^n_+$, что

$$\nabla U(x^0) = \lambda_0 p + \lambda, \tag{1.24}$$

$$\lambda_0(I - p^T x^0) = 0, \tag{1.25}$$

$$\lambda_j x^0_j = 0, \quad j = 1, \ldots, n. \tag{1.26}$$

Из (1.24) и (1.26) следует, что

$$\frac{1}{p_j}\frac{\partial U(x^0)}{\partial x_j} = \lambda_0, \quad \text{для всех } j, \text{ для которых } x^0_j > 0, \tag{1.27}$$

т.е. отношения *предельной полезности* $\frac{\partial U(x^0)}{\partial x_j}$ к цене p_j должно быть одинаковым для всех закупленных товаров j. Считая, что некоторые товары были куплены, из (1.27) следует, что оптимальный множитель λ_0 должен быть положительным. Тогда из (1.25) следует, что весь доход должен быть израсходован: $I - p^T x^0 = 0$.

Будем считать, что потребители покупают все виды товаров и услуг (в противном случае можно уменьшить размерность пространства товаров, исключая из рассмотрения непокупаемые товары). Тогда условия (1.24)–(1.26) примут вид

$$\nabla U(x^0) - \lambda_0 p = 0, \quad I - p^T x^0 = 0. \tag{1.28}$$

Геометрическая иллюстрация условий (1.28) для $n = 2$ приведена на рис. 1.4. Мы видим, что оптимальное решение x^0 задачи потребления является точкой касания бюджетной гиперплоскости $p^T x = I$ с поверхностью безраличия $U(x) = U(x^0)$. Исходя из этого наблюдения, ответьте на следующий вопрос: как изменится набор потребления, если цена продукта 1 увеличится?

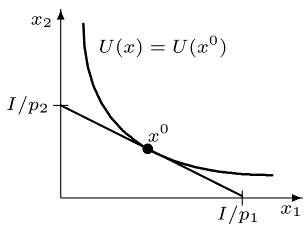


Рис. 1.4.

1.4.2. Модель равновесия Фишера

Рассмотрим рынок с n делимыми *продуктами* и m *потребителями*. Предположим, что на рынке имеется b_j единиц продукта j, а потребитель i обладает суммой денег a_i, и $U_i : \mathbb{R}^n_+ \to \mathbb{R}$ есть его функция полезности.

Говорят, что вектор цен $p = (p_1, \ldots, p_n)^T \in \mathbb{R}^n_+$ *освобождает рынок*, если для оптимальных векторов потребления

$$\bar{x}^i \in \arg\max \left\{ U_i(x) : \sum_{j=1}^{n} p_j x_j \le a_i,\ x \in \mathbb{R}^n_+ \right\}, \quad i = 1, \ldots, m, \tag{1.29}$$

каждый потребитель полностью тратит все свои деньги

$$\sum_{j=1}^{n} p_j \bar{x}^i_j = a_i, \quad i = 1, \ldots, m, \tag{1.30}$$

и все продукты потребляются полностью

$$\sum_{i=1}^{m} x^i_j = b_j\,, \quad j = 1, \ldots, n. \tag{1.31}$$

Как мы скоро увидим, задача поиска равновесия в модели Фишера сводится к решению задачи выпуклого программирования Эйзенберга — Гейла:

$$\sum_{i=1}^{m} a_i \log U_i(x^i) \to \max, \tag{1.32a}$$

$$\sum_{i=1}^{m} x^i_j \le b_j, \quad j = 1, \ldots, n, \tag{1.32b}$$

$$x^i_j \ge 0, \quad j = 1, \ldots, n,\ \ i = 1, \ldots, m, \tag{1.32c}$$

где $x^i \stackrel{\text{def}}{=} (x^i_1, \ldots, x^i_n)^T$ есть вектор переменных, компонента x^i_j которого — это количество продукта j, покупаемое потребителем i.

Напомним, что функция полезности является вогнутой и неубывающей по всем своим аргументам. В силу этого задача (1.32) есть задача максимизации вогнутой функции при линейных ограничениях, для решения которой имеются эффективные алгоритмы выпуклого программирования.

Теорема 1.6. *Если все функции полезностей $U_i(x)$ являются однородными ($U_i(tx) = tU_i(x)$ для всех $x \in \mathbb{R}^n_+$) и непрерывно дифференцируемыми, и для каждого продукта j хотя бы одна из функций $U_i(x)$ строго возрастает по x_j $\left(\frac{\partial U_i(x)}{\partial x_j} > 0\right.$ для всех $\left. x \in \mathbb{R}^n_+\right)$, то в модели Фишера существует равновесие.*

Доказательство. Пусть векторы $\bar{x}^1, \ldots, \bar{x}^n$ образуют оптимальное решение задачи (1.32). Запишем условия оптимальности Куна — Таккера, обозначив множители Куна — Таккера, соответствующие ограничениям (1.32b), через p_j (мы будем интерпретировать p_j как цену продукта j), а ограничениям (1.32c), через μ^i_j:

$$\frac{a_i}{U_i(\bar{x}^i)} \times \frac{\partial U_i(\bar{x}^i)}{\partial x^i_j} = p_j - \mu^i_j, \quad j = 1, \ldots, n, \;\; i = 1, \ldots, m, \tag{1.33a}$$

$$p_j \left(\sum_{i=1}^{m} \bar{x}^i_j - b_j \right) = 0, \quad j = 1, \ldots, n, \tag{1.33b}$$

$$\mu^i_j \bar{x}^i_j = 0, \quad j = 1, \ldots, n, \;\; i = 1, \ldots, m, \tag{1.33c}$$

$$p_j \geq 0, \quad j = 1, \ldots, n, \tag{1.33d}$$

$$\mu^i_j \geq 0, \quad j = 1, \ldots, n, \;\; i = 1, \ldots, m. \tag{1.33e}$$

Для фикфированного i умножим j-е равенство в (1.33a) и затем сложим n равенств. В результате получим равенства

$$\frac{a_i}{U_i(\bar{x}^i)} \sum_{j=1}^{n} \bar{x}^i_j \frac{\partial U_i(\bar{x}^i)}{\partial x^i_j} = \sum_{j=1}^{n} (p_j - \mu^i_j) \bar{x}^i_j \quad i = 1, \ldots, m.$$

Используя формулу Эйлера $U_i(x^i) = \sum_{j=1}^{n} x^i_j \frac{\partial U_i(x^i)}{\partial x^i_j}$, справедливую в силу однородности функции U_i, преобразуем эти равенства в следующие:

$$a_i = \sum_{j=1}^{n} \bar{x}^i_j, \quad i = 1, \ldots, m,$$

которые означают, что все потребители тратят свои деньги полностью.

Пусть x есть произвольный вектор потребления потребителя i:

$$\sum_{j=1}^{n} p_j x_j \leq a_i.$$

Покажем, что $U_i(x) \le U_i(\bar{x}^i)$. В силу вогнутости функции U_i, используя равенства (1.33a), имеем

$$\begin{aligned} U_i(x) - U_i(\bar{x}^i) &\le \left(\nabla U_i(\bar{x}^i)\right)^T (x - \bar{x}^i) \\ &= \frac{U_i(\bar{x}^i)}{a_i} \sum_{j=1}^n (p_j - \mu_j^i)(x_j - \bar{x}_j^i) \\ &= \frac{U_i(\bar{x}^i)}{a_i} \left(\sum_{j=1}^n (p_j x_j - \mu_j^i x_j) - a_i \right) \\ &\le \frac{U_i(\bar{x}^i)}{a_i} \left(\sum_{j=1}^n p_j x_j - a_i \right) \le 0. \end{aligned}$$

Итак, мы показали, что $\bar{x}^i$ — это наилучший набор потребления для потребителя i.

Нам осталось показать, что все продукты потребляются полностью. Из условия (1.33b) следует, что все продукты с ненулевой ценой $p_j > 0$ потребляются полностью: $\sum_{j=1}^n \bar{x}_j^i = b_j$. Поэтому, если предположить, что какой-то продукт j используется не полностью, т. е. $\sum_{i=1}^m \bar{x}_j^i < b_j$, то его цена $p_j = 0$. Пусть i есть тот потребитель, функция полезности которого строго возрастает по переменной x_j^i Тогда потребитель i мог бы увеличить свою полезность, купив за нулевую сумму весь остаток $b_j - \sum_{i=1}^m \bar{x}_j^i$ продукта j. Но это противоречило бы тому, что $\bar{x}^i$ — оптимальный набор потребления для потребителя i. □

В заключение отметим, что условия теоремы 1.6 выполняются для *линейных функций полезности*

$$U_i(x) \stackrel{\text{def}}{=} \sum_{j=1}^n u_j^i x_j, \quad i = 1, \dots, m,$$

где u_j^i есть полезность потребителя i от обладания единицей продукта j, если предположить, что любой продукт j полезен хотя бы для одного потребителя i ($u_j^i > 0$).

1.4.3. Метод максимального правдоподобия

Пусть $X \subseteq \mathbb{R}^n$ и для каждого $x \in X$ задано распределение вероятностей на $\mathbb{R}^m$ с плотностью $p_x : \mathbb{R}^m \to [0, 1]$. Для фиксированного $y \in \mathbb{R}^m$

мы можем рассматривать $p_x(y)$ как функцию аргумента $x \in X$, которую называют *функцией правдоподобия*. На практике удобнее использовать *логарифмическую функцию правдоподобия*: $l(x) = \ln p_x(y)$.

Рассмотрим задачу оценивания вектора параметров x по одному наблюдению случайного вектора y. Один из наиболее часто используемых методов, называемый *методом максимального правдоподобия*, в качестве оценки вектора x вычисляет вектор

$$x^{\mathrm{ML}} \in \arg\max_{x \in X} l(x). \tag{1.34}$$

Линейные измерения с одинаково распределенными независимыми шумами

Рассмотрим модель линейных измерений $y = Ax + v$, где A есть $m \times n$-матрица, $y \in \mathbb{R}^m$ — наблюдаемый вектор, $x \in X \subseteq \mathbb{R}^n$ — вектор оцениваемых параметров, $v \in \mathbb{R}^m$ — вектор ошибок измерений (или шум). Мы предполагаем, что все шумы v_i есть независимые одинаково распределенные случайные величины с плотностью $p : \mathbb{R} \to [0, 1]$. Тогда функция правдоподобия имеет вид:

$$p_x(y) = \prod_{i=1}^{m} p\left(y_i - \sum_{j=1}^{n} a_{ij} x_j\right),$$

Логарифмическая функция правдоподобия записывается следующим образом:

$$l(x) = \ln p_x(y) = \sum_{i=1}^{m} \ln p\left(y_i - \sum_{j=1}^{n} a_{ij} x_j\right),$$

Чтобы оценить вектор параметров x по методу максимального правдоподобия, нужно решить следующую оптимизационную задачу:

$$\max\left\{\sum_{i=1}^{m} \ln p\left(y_i - \sum_{j=1}^{n} a_{ij} x_j\right) : x \in X\right\}. \tag{1.35}$$

Теперь приведем примеры оценивания по методу максимального правдоподобия для некоторых часто используемых распределений.

- *Гаусовый шум.* Когда случайные величины v_i распределены по нормальному закону с матожиданием 0 и дисперсией σ^2, то плотность задается формулой

$$p(z) = \frac{1}{\sqrt{2\pi}\,\sigma} \mathrm{e}^{-z^2/2\sigma^2}.$$

Логарифмическая функция правдоподобия, определенная на $X = \mathbb{R}^n$, имеет вид:

$$l(x) = -(m/2)\ln(2\pi\sigma^2) - \frac{1}{\sigma^2}\sum_{i=1}^{m}\left(y_i - \sum_{j=1}^{n} a_{ij}x_j\right)^2 =$$
$$= -(m/2)\ln(2\pi\sigma^2) - \frac{1}{\sigma^2}\|Ax - y\|_2^2.$$

Поэтому оценкой x по методу максимального правдоподобия будет вектор

$$x^{ML} \in \arg\min \|Ax - y\|_2.$$

- *Лапласовый шум.* Когда случайные величины v_i распределены по экспоненциальному закону с плотностью

$$p(z) = \frac{1}{2a} \mathrm{e}^{-|z|/a},$$

где $a > 0$, то логарифмическая функция правдоподобия, определенная на $X = \mathbb{R}^n$, имеет вид:

$$l(x) = -m\ln(2a) - \frac{1}{a}\sum_{i=1}^{m}\left|y_i - \sum_{j=1}^{n} a_{ij}x_j\right| =$$
$$= -m\ln(2a) - \frac{1}{a}\|Ax - y\|_1.$$

Поэтому оценкой x по методу максимального правдоподобия будет вектор

$$x^{ML} = \arg\min \|Ax - y\|_1.$$

- *Однородный шум.* Когда случайные величины v_i равномерно распределены на отрезке $[-a, a]$, то плотность распределения вероятностей

$$p(z) = \frac{1}{2a} \quad \text{для } z \in [-a, a].$$

Логарифмическая функция

$$l(x) = -m\ln(2a)$$

постоянна для всех $x \in X = [-a, a]^n$. Поэтому оценкой x по методу максимального правдоподобия будет любой вектор x, который уовлетворяет неравенствам

$$\left| y_i - \sum_{j=1}^{n} a_{ij} x_j \right| \le a, \quad i = 1, \dots, m.$$

Логистическая регрессия

Рассмотрим случайную величину $y \in \{0, 1\}$ с

$$\mathsf{P}(y = 1) = p, \quad \mathsf{P}(y = 0) = 1 - p \quad (0 \le p \le 1).$$

Предполагается, что вероятность p зависит от объясняющих переменных $u \in \mathbb{R}^n$. Например, $y = 1$ может означать, что индивидуум в некоторой популяции страдает некоторым заболеванием. Вероятность p обнаружения болезни есть функция некоторых объясняющих переменных u, которые могут представлять возвраст, вес, рост, кровяное давление и другие медицинские показатели.

Логистическая модель имеет вид

$$p = \frac{\exp(a^T u + b)}{1 + \exp(a^T u + b)}, \tag{1.36}$$

где $a \in \mathbb{R}^n$ и $b \in \mathbb{R}$ есть параметры модели, которые нужно определить.

Исходными данными для определения параметров a и b являются пары (u^i, y_i) $(i = 1, \dots, m)$, где $y_i \in \{0, 1\}$ — это значение величины y, когда вектор u объясняющих переменные принял значение $u^i \in \mathbb{R}^n$. Параметры a и b определим по методу максимального правдоподобия. В таком случае логистическую модель также называют *логистической регрессией.*

Предположим, что исходные данные упорядочены таким образом, что $y_1 = \dots = y_k = 1$, а $y_{k+1} = \dots = y_m = 0$. Тогда функция максимального правдоподобия записывается следующим образом:

$$p_{a,b}(y_1, \dots, y_m) = \left(\prod_{i=1}^{k} p_i \right) \times \left(\prod_{i=k+1}^{m} (1 - p_i) \right),$$

где

$$p_i = \frac{\exp(a^T u^i + b)}{1 + \exp(a^T u^i + b)}, \quad i = 1, \dots, m.$$

Логарифмическая функция максимального правдоподобия имеет вид

$$\begin{aligned} l(a,b) &= \sum_{i=1}^{k} \ln p_i + \sum_{i=k+1}^{m} \ln(1-p_i) = \\ &= \sum_{i=1}^{k} \ln \frac{\exp(a^T u^i + b)}{1+\exp(a^T u^i + b)} + \sum_{i=k+1}^{m} \ln \frac{1}{1+\exp(a^T u^i + b)} = \\ &= \sum_{i=1}^{k} (a^T u^i + b) - \sum_{i=1}^{m} \ln(1+\exp(a^T u^i + b)). \end{aligned}$$

Поскольку функция $l(a,b)$ вогнута по переменным a и b, то задача построения логистической регрессии есть задача максимизации вогнутой функции, для решения которой существуют эффективные алгоритмы. Отметим также, что на практике в каждом конкретном случае возможны дополнительне ограничения на парметры a и b. Например, в задаче оценивания вероятности обнаружения болезни, если u_i есть возраст пациента, то логично потребовать, чтобы коэффициент a_i был неотрицательным, поскольку с возрастом вероятность заболевания увеличивается.

1.5. Мультикритериальные задачи

Очень часто при решении той или иной практической задачи мы стремимся достичь сразу несколько целей. Как правило, эти цели противоречат друг другу. Например, проектируя самолет, мы хотели бы одновременно увеличить его скорость и грузоподёмность.

Ради простоты изложения, мы здесь ограничимся рассмотрением задачи *многокритериальной оптимизации* в следуюшей постановке:

$$\min\{f(x) :\ x \in X\}, \tag{1.37}$$

где $X \subseteq \mathbb{R}^n$, а $f : X \to \mathbb{R}^m$ есть m-мерная вектор-функция, которая в точке $x \in X$ принимает сразу m значений $f(x) = (f_1(x), \ldots, f_m(x))^T$. Чтобы сделать постановку задачи (1.37) содержательной, нам нужно определиться с тем, что является минимумом векторной функции.

Для пары допустимых решений $x, y \in X$ разделим наши m критериев на три группы $L(x,y)$, $G(x,y)$ и $E(x,y)$:

$$\begin{aligned} f_i(x) &\le f_i(y), \quad i \in L(x,y),\\ f_i(x) &\ge f_i(y), \quad i \in G(x,y),\\ f_i(x) &= f_i(y), \quad i \in E(x,y). \end{aligned}$$

Здесь $L(x,y)$ есть множество критериев, для которых решение x лучше решения y, $G(x,y)$ есть множество критериев, для которых решение x хуже решения y, а $E(x,y)$ — это множество критериев, относительно которых решения x и y равноценны. Если $E(x,y) = \{1,\dots,m\}$, то решения x и y равноценны. Если $G(x,y) = \emptyset$, то говорят, что решение x не *хуже по Паретто*, чем решение y. Если $G(x,y) = \emptyset$ и $L(x,y) \neq \emptyset$, то решение x *лучше по Паретто*, чем решение y. В случае, когда $G(x,y) \neq \emptyset$ и $L(x,y) \neq \emptyset$, то говорят, что решения x и y *несравнимы по Паретто*.

Решение $x \in X$ называется *оптимальным по Паретто* для задачи (1.37), если в X нет другого решения, которое лучше по Паретто, чем решение x. Теперь мы можем сказать, что целью в задаче (1.37) является поиск всех оптимальных по Паретто решений. Но во многих случаях эту цель осуществить очень трудно, а иногда и невозможно, из-за огромного числа оптимальных по Паретто решений. К тому же, при наличии нескольких решений нам все равно нужно будет выбрать одно из них для реализации на практике. Поэтому на практике изначально ставится более скромная задача — найти одно оптимальное по Паретто решение задачи (1.37), которое или 1) оптимально для некоторого скалярного критерия, или 2) является лексикографически оптимальным (или «почти» лексикографически оптимальным) для некоторого упорядочения критериев $f_1,\dots,f_m$.

1.5.1. Скаляризация векторного критерия

Среди способов скаляризации векторных критериев на практике наиболее часто используются два способа: свертка критериев и целевое программирование. Свертку критериев следует использовать тогда, когда значения всех критериев можно выразить в одной единице измерения. Когда в задаче имеются критерии, которые измеряются в разных единицах, то содержательный смысл свертки (взвешенной суммы) таких критериев непонятен (нельзя приписать какой-либо смысл сумме килограмма и секунды). В подобных случаях используют метод, который называют «целевым программированием», суть которого в том, чтобы найти такое решение, для которого значения всех критериев близки к заранее заданным целевым значениям.

Свертка критериев

Свертка критериев — это один из стандартных способов найти оптимальное по Паретто решение в задаче мультикритериальной оптимизации (1.37). Каждому критерию $i = 1, \dots, m$ приписывается некоторый вес $\lambda_i \geq 0$ и затем решается оптимизационная задача

$$\min\{\lambda^T f(x) :\ x \in X\} \tag{1.38}$$

с одним критерием, который есть взвешенная сумма $\sum_{i=1}^m \lambda_i f_i(x)$ критериев $f_1, \dots, f_m$.

Если все веса λ_i положительны, то оптимальное решение x^0 задачи (1.38) является оптимальным по Паретто для задачи (1.37). Действительно, если бы это было не так, то существовала бы точка $x^1 \in X$, которая лучше x^0: $f_i(x^1) \leq f_i(x^0)$ для всех $i = 1, \dots, m$, и $f_{i_0}(x^1) < f_{i_0}(x^0)$ для некотрого i_0. Складывая неравенства $\lambda_i f_i(x^1) \leq \lambda_i f_i(x^0)$, для $i = 0, \dots, m$, получим неравенство

$$\sum_{i=1}^m \lambda_i f_i(x^1) < \sum_{i=1}^m \lambda_i f_i(x^0),$$

которое противоречит тому, что x^0 есть оптимальное решение задачи (1.38).

Но верно ли обратное:

> можно ли подобрать веса таким образом, чтобы оптимальным в задаче (1.38) оказалось любое заданное оптимальное по Паретто решение задачи (1.37)?

В общем случае ответ на этот вопрос отрицательный. На рис. 1.5 изображена область допустимых значений $f(X) \stackrel{\text{def}}{=} \{f(x) = (f_1(x), f_2(x))^T : x \in X\}$ двухкритериальной задачи оптимизации. Множество оптимальных по Паретто решений является частью границы (на рисунке она изображена жирной линией) области $f(X)$. На данном рисунке также изображены точки $f(x^1)$, $f(x^2)$ и $f(x^3)$ для трех оптимальных по Паретто оптимальных решений x^1, x^2 и x^3. Множество $f(X)$ лежит по одну сторону от касательных к его границе в точках $f(x^1)$ и $f(x^3)$. Если в качестве векторов весов λ^1 и λ^3 взять векторы нормалей к этим касательным,

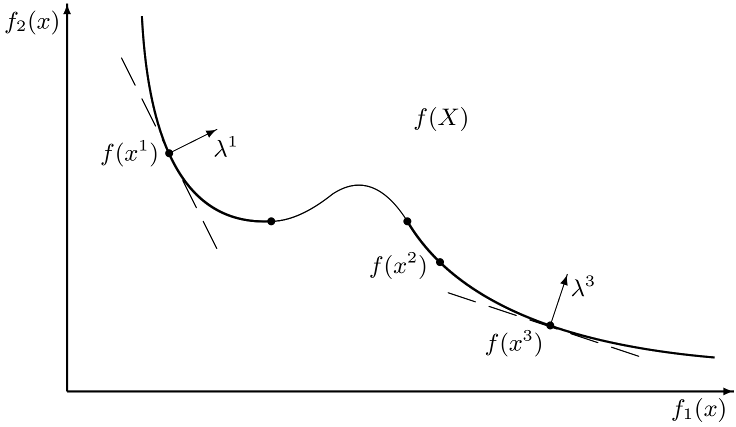


Рис. 1.5. Геометрическая интерпретация свертки критериев

то точка x^1 будет решением задачи (1.38) при $\lambda = \lambda^1$, а точка x^2 будет решением задачи (1.38) при $\lambda = \lambda^2$.

Поскольку любая прямая

$$\lambda_1 f_1(x) + \lambda_1 f_2(x) = b(\lambda_1, \lambda_2) \stackrel{\text{def}}{=} \lambda_1 f_1(x^2) + \lambda_1 f_2(x^2),$$

проходящая через точку $f(x^2) = (f_1(x^2), f_2(x^2))$ делит множество $f(X)$ на два непустых множества

$$X_1 = \{x \in X : \ \lambda_1 f_1(x) + \lambda_2 f_2(x) \le b(\lambda_1, \lambda_2)\},$$
$$X_2 = \{x \in X : \ \lambda_1 f_1(x) + \lambda_2 f_2(x) > b(\lambda_1, \lambda_2)\},$$

то точка x^2 не может быть оптимальным решением задачи (1.38) ни при каких весах λ_1 и λ_2, одновременно не равных нулю.

Для важного частного случая задачи (1.37), когда X — выпуклое множество и все критерии $f_1, \ldots, f_m$ — выпуклые на X функции ответ на поставленый выше вопрос утвердительный. Действительно, если $x^0 \in X$ есть оптимальное по Паретто решение задачи (1.37), то $f(x^0)$ является граничной точкой множества $f(X)$, и по теореме об отделении выпуклых множеств, существует гиперплоскость $a^T y = b$, такая, что $a^T f(x) \ge b$ для всех $x \in X$ и $a^T f(x^0) \le b$. Последнее означает, что для $\lambda = a$ точка x^0 является оптимальным решением задачи (1.38).

Целевое программирование

Предположим, что нам известны *целевые* значения $g_1, \ldots, g_m$ для всех критериев $f_1, \ldots, f_m$, отклонения от которых в большую сторону нежелательны. Например, мы хотели бы разместить несколько дополнительных станций скорой помощи, чтобы достичь следующих целей:

1) среднее время отклика (от звонка больного до момента прибытия к нему скорой помощи) не должно превосходить 5 минут;
- количество потенциальных больных, которые не смогут получить помощь в течении 10 минут, не должно превосходить 10 процентов от их общего количества;
3) как можно меньше отклониться от бюджета в 250 тыс. долларов.

В целевом программировании задача многокритериальной оптимизации (1.37) заменяется следующей оптимизационной задачей с одним скадярным критерием:

$$\begin{aligned}
&\sum_{i=1}^{m} w_i h_i(s_i) \to \min, \\
&f_i(x) - s_i \le g_i, \quad i = 1, \ldots, m, \\
&s \in \mathbb{R}_+^m, \ x \in X,
\end{aligned} \tag{1.39}$$

где $s = (s_1, \ldots, s_m)^m$ есть вектор (дополнительных) переменных *избытка*, а $h_i(s_i)$ — это штраф за превышением критерием i его целевого значения на величину s_i.

Целью в задаче (1.39) является минимизация суммы штрафов за отклонение компонент векторного китерия от их целевых значений. При естественном предположении, что все функции штрафов являются возрастающими, для оптимального решения (x^0, s^0) задачи (1.39) справедливы равенства: $s_i^0 = \min\{0, f_i(x^0) - g_i\}$, $i = 1, \ldots, m$. Это означает, что штрафуются только отклонения критериев от их целевых значений в большую сторону. При выборе весов w_i мы должны учитывать значимость критериев. Если предположить, что критерии изначально пронумерованы в порядке их значимости, то веса должны удовлетворять условию: $0 < w_1 \le w_2 \le \cdots \le w_m$.

На практике наиболее часто используются *линейные* $h_i(s_i) = s_i$ и *квадратичные* $h_i(s_i) = s_i^2$ штрафные функции.

1.5.2. Лексикографическая оптимизация

Основное затруднение при решение многокритериальных задач состоит в том, что мы не можем сравнивать любые векторы из $\mathbb{R}^m$. Простое покоординатное сравнение векторов ($u \le v$, если $u_i \le v_i$ для $i = 1, \dots, m$) определяет только *частичный порядок*. Так, оно не позволяет сравнить двумерные векторы $(2, 1)^T$ и $(1, 2)^T$. Мы можем определить *полный* (или *линейный*) порядок на $\mathbb{R}^m$ разными способами. В контексте многокритериальной оптимизации наиболее часто используется *лексикографический порядок*.

Рассмотрим две точки $u, v \in \mathbb{R}^m$. Говорят, что u *лексикографически меньше* чем v и записывается $u \prec_{\text{lex}} v$, если для некоторого k, $1 \le k < m$, выполняются условия $u_i = v_i$ для $i = 1, \dots, k-1$ и $u_k < v_k$. Обозначение $u \preceq_{\text{lex}} v$ означает, что $u \prec_{\text{lex}} v$ или $u = v$.

Пусть $X \subseteq \mathbb{R}^n$ и $f : X \to \mathbb{R}^m$ есть m-мерная вектор-функция. Предположим, что критерии $f_1, \dots, f_m$ пронумерованы с учетом их значимости, т. е. критерий f_i важнее всех последующих критериев $f_{i+1}, \dots, f_m$. В задаче *лексикографической оптимизации*

$$\operatorname{lexmin}\{f(x) :\ x \in X\} \tag{1.40}$$

нужно найти такую точку $x^0 \in X$, что $f(x^0) \preceq_{\text{lex}} f(x)$ для всех $x \in X$. Точка x^0 называется еще *точкой лексикографического минимума*. Заметим, что все точки лексикографического минимума являются оптимальными по Паретто для задачи многокритериальной оптимизации (1.37).

Для $\epsilon \in \mathbb{R}^m_+$ точка $\tilde{x}$ называется *лексикографически ϵ-оптимальным решением* задачи (1.40), если $f(\tilde{x}) - f(x^0) \preceq_{\text{lex}} \epsilon$, где x^0 лексикографический минимум в задаче (1.40). Понятно, что лексикографически 0-оптимальные решения задачи (1.40) являются ее лексикографическими минимумами.

Мы можем найти лексикографически ϵ-оптимальное решение задачи (1.40), для $k = 1, \dots, m$ последовательно решив m оптимизационных задач:

$$\begin{aligned} f_k^* &= \min f_k(x), \\ f_i(x) &\le f_i^* + \epsilon_i, \quad i = 1, \dots, k-1, \\ x &\in X. \end{aligned} \tag{1.41}$$

Решение $\tilde{x}$ последней задачи (при $k = m$) и будет лексикографически ϵ-оптимальным решением задачи (1.40).

1.6. Упражнения

1.1. На параболе $y = x^2 - 2x + 1$ найти точку $(x^0, y^0)^T$, ближайшую к точке $(1, 1)^T$.

1.2. Рассмотрим следующую оптимизационную задачу

$$\begin{aligned} (x_1 - 2/3)(x_2 - 1/2)(x_3 - 1/3) &\to \min, \\ x_1^2 + x_2^2 + x_3^2 &\le 1, \\ x_1 \ge 0,\ x_2 \ge 0, x_3 &\ge 0. \end{aligned}$$

а) Удовлетворяют ли точки $x^1 = (0, 0, 0)^T$ и $x^2 = (0, 0, 1)^T$ условиям Куна-Таккера;

б) Какие из точек x^1 и x^2 являются локальными минимумами рассматриваемой задачи?

1.3. Рассмотрим следующую оптимизационную задачу

$$\begin{aligned} \mathrm{e}^{x_1 + x_3} + x_2^4 + 4x_2 - (x_1 + x_3) &\to \min, \\ x_1^2 + x_2^2 + x_3^2 &= 1, \\ x_1^2 - x_2 &= 1. \end{aligned}$$

а) Найдите 3 допустимых решения (допустимых точки) с $x_3 = 0$;

б) Какие из этих 3-х точек удовлетворяют условиям Куна-Таккера, а какие являются локальными минимумами рассматриваемой задачи?

1.4. Предприятие производит два продукта. Функция прибыли для плана $x = (x_1, x_2)^T$ нелинейна (за счет насыщения):

$$f(x) = (4x_1 + 3x_2)\,\mathrm{e}^{-(x_1 + x_2)/500}.$$

За плановый период предприятие может произвести не более 300 единиц продукта 1 и не более 400 единиц продукта 2.

Найти оптимальный (для которого прибыль максимальна) план производства $x^* = (x_1^*, x_2^*)^T$.

1.5. Потребитель хочет потратить \$100 на покупку двух делимых продуктов. Стоимость единицы продукта 1 равна \$2, а продукта 2 — \$3. Сколько единиц x_1 и x_2 продуктов 1 и 2 должен купить потребитель, чтобы максимизировать свою функцию полезности $U(x_1, x_2) = 2\ln(x_1) + 3\ln(x_2)$.

1.6. Два индивидуума потребляют только два продукта. Функция полезности индивидуума i $(i = 1, 2)$ имеет вид $u_i = u_i(x_{i1}, x_{i2}) \overset{\mathrm{def}}{=} x_{i1}^{\sigma_i} x_{i2}^{1-\sigma_i}$, где x_{ij} есть количество продукта j $(j = 1, 2)$, потребляемое i-м индивидуумом. В наличии имеется a_j единиц продукта j, $j = 1, 2$.

Нужно найти распределение продуктов между индивидуумами

$$x_{11}^*, x_{12}^*, x_{21}^*, x_{22}^*,$$

для которого функция благосостояния

$$W(x_{11}, x_{12}, x_{21}, x_{22}) = u_1(x_{11}, x_{12}) \cdot u_2(x_{21}, x_{22}) = x_{11}^{\sigma_1} x_{12}^{1-\sigma_1} x_{21}^{\sigma_2} x_{22}^{1-\sigma_2}$$

принимает максимальное значение. Дайте интерпретацию множителям Куна — Таккера. (Предполагается, что $0 < \sigma_1, \sigma_2 < 1$.)

1.7. Решите следующую оптимизационную задачу

$$\begin{aligned} (x_1 - 2)^2 + (x_2 - 2)^2 &\to \min, \\ x_1^2 + x_2^2 &\le 1, \\ x_1 + x_2 &\le a, \\ x_1,\ x_2 &\ge 0 \end{aligned}$$

при всех значениях параметра a.

Глава 2
Линейное программирование

Задача линейного программирования (ЛП) есть задача максимизации линейной функции при линейных ограничениях. Задачу ЛП можно записать несколькими стандартными способами. Мы здесь рассмотрим только три таких способа.

Задача ЛП в канонической форме записывается следующим образом:

$$\max\{c^T x : \ Ax \le b\}, \tag{2.1}$$

где A — действительная матрица размера $m \times n$, $c \in \mathbb{R}^n$, $b \in \mathbb{R}^m$, а $x = (x_1, \dots, x_n)^T$ есть вектор неизвестных. В дальнейшем мы будем предполагать, что матрица ограничений задачи (2.1) имеет полный столбцовый ранг, т. е. $\operatorname{rank} A = n$.

Задача ЛП в стандартной форме имеет следующий вид:

$$\max\{c^T x : \ Ax = b, \ x \ge 0\}, \tag{2.2}$$

где A, c, b и x определяются также, как и для задачи ЛП в канонической форме. Для задачи ЛП в стандартной форме обычно предполагается, что A есть матрица полного столбцового ранга, т. е. $\operatorname{rank} A = n$.

Еще одна часто встречающаяся форма задачи ЛП следующая:

$$\max\{c^T x : \ Ax \le b, \ x \ge 0\}, \tag{2.3}$$

где A, c, b и x определяются как и ранее, но здесь не накладывают никаких ограничений на ранг матрицы A.

Эквивалентность задач ЛП в различных формах

Все три задачи, (2.1), (2.2) и (2.3), эквивалентны в том смысле, что любую из них можно преобразовать к форме любой другой задачи.

(2.1)→(2.3). Представим вектор $x = x^+ - x^-$ как разность двух неотрицательных векторов $x^+, x^- \in \mathbb{R}^n_+$. Вводя обозначения

$$\bar{x} = \begin{pmatrix} x^+ \\ x^- \end{pmatrix}, \quad \bar{c} = \begin{pmatrix} c \\ -c \end{pmatrix}, \quad \bar{A} = [A \,|\, -A],$$

запишем задачу (2.1) в форме (2.3):

$$\max\{\bar{c}^T \bar{x} : \bar{A}\bar{x} \le b,\ \bar{x} \ge 0\}.$$

(2.3)→(2.2). Введем вектор $s = (s_1, \dots, s_m)^T$ переменных недостатка. Вводя обозначения

$$\bar{x} = \begin{pmatrix} x \\ s \end{pmatrix}, \quad \bar{c} = \begin{pmatrix} c \\ \mathbf{0} \end{pmatrix}, \quad \bar{A} = [A|I],$$

запишем (2.3) в форме (2.2):

$$\max\{\bar{c}^T \bar{x} : \bar{A}\bar{x} = b,\ \bar{x} \ge 0\}.$$

(2.2)→(2.1). Вводя обозначения

$$\bar{b} = \begin{pmatrix} b \\ -b \\ \mathbf{0} \end{pmatrix}, \quad \bar{A} = \begin{bmatrix} A \\ -A \\ -I \end{bmatrix},$$

запишем (2.2) в форме (2.1):

$$\max\{c^T x : \bar{A}x \le \bar{b}\}.$$

Задача дробно-линейного программирования

Задача дробно-линейного программирования — это задача минимизации дробно-линейной целевой функции при линейных ограничениях:

$$\begin{aligned} \frac{c^T x + d}{u^T x + v} &\to \min, \\ Ax &\le b, \\ Gx &\le h, \\ u^T x + v &> 0. \end{aligned} \tag{2.4}$$

Понятно, что задача (2.4) обобщает задачу ЛП: если $u = 0$ и $v = 1$, то задача (2.4) превращается в задачу ЛП. С другой стороны, задачу (2.4) можно преобразовать в задачу ЛП:

$$\begin{aligned} c^T x + dt &\to \min, \\ Ax - bt &= 0, \\ Gx - ht &= 0, \\ u^T y + vt &= 1, \\ t &\geq 0, \end{aligned} \tag{2.5}$$

с переменными $y = (y_1, \ldots, y_n)^T$ и t.

Докажем эквивалентность задач (2.4) и (2.5), Если x есть допустимое решение задачи (2.4), то пара

$$(y, t) = \frac{1}{u^T x + v}(x, 1)$$

есть допустимое решение задачи (2.5), причем значения целевых функций для обоих решений одинаково: $(c^T x + d)/(u^T x + v) = c^T y + vt$. Из сказанного следует, что оптимальное значение целевой функции в задаче (2.5) не больше оптимального значения целевой функции в задаче (2.4).

Обратно, если (y, t) есть допустимое решение задачи (2.5) с $t > 0$, то $x = y/t$ есть допустимое решение задачи (2.4), причем $(c^T x + d)/(u^T x + v) = c^T y + vt$. Если $(y, 0)$ — допустимое решение задачи (2.5), и $\bar{x}$ — допустимое решение задачи (2.4), то $x(\delta) = \bar{x} + \delta y$ является допустимым решением задачи (2.4) при любых $\delta \geq 0$. Более того, с учетом того, что $u^T y = 1$, имеем

$$\lim_{\delta \to \infty} \frac{c^T x(\delta) + d}{u^T x(\delta) + v} = \lim_{\delta \to \infty} \frac{\left(c^T \bar{x} + d\right) + \delta c^T y}{(u^T \bar{x} + v) + \delta u^T y} = c^T y + d \cdot 0.$$

Таким образом, мы можем найти допустимое решение задачи (2.4) со значением целевой функции сколь угодно близким к значению целевой функции для решения $(y, 0)$ задачи (2.5). Это означает, что значение целевой функции в задаче (2.5) не меньше оптимального значения целевой функции в задаче (2.4).

2.1. Двойственность в линейном программировании

Теория двойственности линейного программирования имеет прямое отношение к проблеме оценки эффективности использования ресурсов в производственных процессах.

Рассмотрим пару двойственныз задач ЛП

$$\max\{c^T x \,:\, Ax \le b,\ x \ge 0\} \tag{П}$$

и

$$\min\{b^T y \,:\, A^T y \ge c,\ y \ge 0\}, \tag{Д}$$

где $c, x \in \mathbb{R}^n$, $b, y \in \mathbb{R}^m$, а A есть действительная матрица размера $m \times n$. Задачы (П) и (Д) будем называть, соответственно, *прямой* и *двойственной* задачами. В отношении к прямой задаче (П) переменные x_j $(j = 1, \ldots, n)$ называются *прямыми*, а переменные y_i $(i = 1, \ldots, m)$ —*двойственными*. Отметим также, что отношение двойственности симметрично, т. е. задача двойственная к двойственной является прямой.

Теорема 2.1 (двойственности). *Имеют место следующие альтернативы.*

1. *Обе задачи (П) и (Д) имееют допустимые решения и*

$$\begin{aligned} &\max\{c^T x \,:\, Ax \le b,\ x \ge 0\} = \\ &\min\{b^T y \,:\, A^T y \ge c,\ y \ge 0\}. \end{aligned} \tag{2.8}$$

2. *Если одна из задач (П) или (Д) не имеет допустимых решений, а другая имеет, то целевая функция этой задачи неограничена.*
3. *Обе задачи (П) и (Д) не имеют допустимых решений.*

Допустимые решения x^* и y^* соответственно задач (П) и (Д) является их оптимальными решениями тогда и только тогда, когда имеют место следующие условия дополняющей нежесткости:

$$(b - Ax^*)^T y^* = 0 \quad \text{и} \quad (c - A^T y^*)^T x^* = 0. \tag{2.9}$$

Доказательство. Сначала докажем утверждение 1. Для конкретности, предположим что задача (П) имеет оптимальное решение x^*. Покажем, что тогда и задача (Д) также имеет оптимальное решение y^*, причем, $c^T x^* = b^T y^*$.

Представим прямую задачу (П) в следующем виде:

$$\begin{aligned} &f(x) = -\sum_{j=1}^{n} c_j x_j \to \min, \\ &g_i(x) = \sum_{j=1}^{n} a_{ij} x_j - b_i \le 0, \quad i = 1, \dots, m, \\ &g_{m+j} = -x_j \le 0, \quad j = 1, \dots n. \end{aligned} \tag{П'}$$

Тогда $\nabla f(x) = -c$, $\nabla g_i(x) = (a_{i1}, \dots, a_{in})^T$ $(i = 1, \dots, m)$, $\nabla g_{m+j}(x) = \mathrm{e}_j$ $(j = 1, \dots, n)$.

Поскольку задача ЛП — это частный случай задачи выпуклого программирования, то по теореме 1.5 существует вектор $\lambda^* \in \mathbb{R}^{m+m}$, что выполняются условия Куна — Таккера:

$$-c_j + \sum_{i=1}^{m} a_{ij}\lambda_i^* + \lambda_{m+j}^* = 0, \quad j = 1, \dots, n, \tag{2.11a}$$

$$\lambda_i^* \left(\sum_{j=1}^{n} a_{ij} x_j^* - b_i \right) = 0, \quad i = 1, \dots, m, \tag{2.11b}$$

$$\lambda_{m+j}^* x_j^* = 0, \quad j = 1, \dots, n, \tag{2.11c}$$

$$\lambda_j^* \ge 0, \quad j = 1, \dots, m+n. \tag{2.11d}$$

Из (2.11) имеем, что вектор $y^* = (y_1^* = \lambda_1^*, \dots, y_m^* = \lambda_m^*)^T$ удовлетворяет следующим условиям:

$$\sum_{i=1}^{m} a_{ij} y_i^* \ge c_j, \quad j = 1, \dots, n, \tag{2.12a}$$

$$y_i^* \left(\sum_{j=1}^{n} a_{ij} x_j^* - b_i \right) = 0, \quad i = 1, \dots, m, \tag{2.12b}$$

$$x_j^* \left(\sum_{i=1}^{m} a_{ij} y_i^* - c_j \right) = 0, \quad j = 1, \dots, n. \tag{2.12c}$$

Заметим, что равенства (2.12c) вытекают из условий (2.11a) и (2.11c).

Из (2.11d) и (2.12a) следует, что точка y^* является допустимым решением двойственной задачи (Д). Докажем, что y^* — оптимальное решение задачи (Д).

Для произвольных допустимых решений $x \in \mathbb{R}^n$ и $y \in \mathbb{R}^m$ соответственно задач (П) и (Д) имеем

$$\sum_{j=1}^{n} c_j x_j \le \sum_{j=1}^{n} \left(\sum_{i=1}^{m} a_{ij} y_i \right) x_j = \sum_{i=1}^{m} \left(\sum_{j=1}^{n} a_{ij} x_j \right) y_i \le \sum_{i=1}^{m} b_i y_i.$$

Это означает, что оптимальное значение целевой функции в прямой задаче (П) не может превосходить оптимальное значение целевой функции в двойственной задаче (Д).

С другой стороны, складывая m равенств в (2.12b), имеем

$$\sum_{i=1}^{m} b_i y_i^* = \sum_{i=1}^{m} \left(\sum_{j=1}^{n} a_{ij} x_j^* \right) y_i^* = \sum_{j=1}^{n} \left(\sum_{i=1}^{m} a_{ij} y_i^* \right) x_j^* \le \sum_{j=1}^{n} c_j x_j^*.$$

Полученное неравенство доказывает, что y^* есть оптимальное решение задачи (Д).

Докажем теперь утверждение 2. Для конкретности предположим, что задача (П) не имеет допустимых решений, а задача (Д) имеет допустимое решение $\bar{y}$. Поскольку система неравенств $Ax \le b$, $x \ge 0$ не имеет решения, то по лемме Фаркаша (лемма A.1) существует точка $(\tilde{y}, s) \in \mathbb{R}^m_+ \times \mathbb{R}^n_+$, что $A^T\tilde{y} + s = 0$ и $b^T\tilde{y} < 0$. Но тогда для $y(\lambda) \stackrel{\text{def}}{=} \bar{y} + \lambda\tilde{y}$ имеем

$$\begin{aligned} A^T y(\lambda) &= A^T\bar{y} + \lambda A^T\tilde{y} \ge c + s \ge c, \quad \text{для всех } \lambda > 0, \\ \lim_{\lambda\to+\infty} b^T y(\lambda) &= b^T\bar{y} + \lim_{\lambda\to+\infty} \lambda b^T\tilde{y} = -\infty. \end{aligned}$$

Это означает, что целевая функция задачи (Д) неограниченно убывает вдоль луча $\{y(\lambda) : \lambda \in \mathbb{R}_+\}$ с началом в точке $\bar{y}$.

Чтобы доказать утверждение 3, достаточно привести пример пары двойственных задач ЛП, каждая из которых не имеет решения:

$$\max\{-x : 0x \le -1\}, \quad \min\{-y : 0y = -1, \ y \ge 0\}.$$

Нам осталось доказать вторую часть теоремы. Пусть x^* и y^* допустимые решения соответственно задач (П) и (Д), для которых выполняются условия дополняющей нежесткости. Тогда

$$c^T x^* = (y^*)^T A x^* = b^T y^*$$

и из доказательства утверждения 1 следует, что точки x^* и y^* являются оптимальными решениями соответственно задач (П) и (Д).

Таблица 2.1

Пара двойственных задач ЛП

Прямая задача	Двойственная задача
$\max c^T x$	$\min b^T y$
$A_i x \leq b_i\,,\ i \in \mathcal{R}_1$	$y_i \geq 0\,,\ i \in \mathcal{R}_1$
$A_i x = b_i\,,\ i \in \mathcal{R}_2$	$y_i \in \mathbb{R}\,,\ i \in \mathcal{R}_2$
$A_i x \geq b_i\,,\ i \in \mathcal{R}_3$	$y_i \leq 0\,,\ i \in \mathcal{R}_3$
$x_j \geq 0\,,\ j \in \mathcal{C}_1$	$y^T A^j \geq c_j\,,\ j \in \mathcal{C}_1$
$x_j \in \mathbb{R}\,,\ j \in \mathcal{C}_2$	$y^T A^j = c_j\,,\ j \in \mathcal{C}_2$
$x_j \leq 0\,,\ j \in \mathcal{C}_3$	$y^T A^j \leq c_j\,,\ j \in \mathcal{C}_3$

Если x^* и y^* есть оптимальные решения соответственно задач (П) и (Д), то

$$c^T x^* = b^* T y^* \geq (y^*)^T A x^* \geq c^T x^*.$$

Откуда $b^* T y^* = (y^*)^T A x^*$ или $(y^*)(b - Ax^*) = 0$.

Аналогично, из

$$b^* T y^* = c^T x^* \leq (y^*)^T A x^* \leq (y^*) T b$$

имеем $c^T x^* = (y^*)^T A x^*$ или $(c - A^T y^*)^T x^* = 0$. □

Общее правило для записи двойственной задачи для данной задачи ЛП приведено в табл. 2.1. Например, двойственной для следующей задачи ЛП

$$\begin{array}{rcrcrcl}
2x_1 & - & 4x_2 & + & 3x_3 & \to & \max, \\
x_1 & + & x_2 & - & x_3 & = & 9, \\
-2x_1 & + & x_2 & & & \leq & 5, \\
x_1 & & & - & 3x_3 & \geq & 4, \\
x_1 & & & & & \geq & 0, \\
 & & & & x_3 & \leq & 0
\end{array}$$

будет задача

$$\begin{array}{rcrcrcr}
9y_1 & + & 5y_2 & + & 4y_3 & \to & \min, \\
y_1 & - & 2y_2 & + & y_3 & \geq & 2, \\
y_1 & + & y_2 & & & = & -4, \\
-y_1 & & & - & 3y_3 & \leq & 3, \\
 & & y_2 & & & \geq & 0, \\
 & & & & y_3 & \leq & 0.
\end{array}$$

2.1.1. Двойственные переменные и теневые цены

Предприятие планирует произволить n видов продукции, используя m видов ресурсов: для производства единицы j-го продукта требуется a_{ij} единиц i-го ресурса. Стоимость единицы j-го продукта равна c_j. В наличии имеется b_i единиц i-го ресурса. Нужно определить план производства, с целью максимизировать прибыль. Обозначив через x_j объем выпуска продукции j-го вида ($j = 1, \dots, n$), мы можем записать задачу поиска оптимального производственного плана следующим образом:

$$\sum_{j=1}^{n} c_j x_j \to \max,$$
$$\sum_{j=1}^{n} a_{ij} \le b_i, \quad i = 1, \dots, m,$$
$$x_j \ge 0, \quad j = 1, \dots, n,$$

или в матричном виде

$$z(b) \stackrel{\text{def}}{=} \max\{c^T x \,:\, Ax \le b,\; x \ge 0\}. \tag{2.13}$$

Пусть x^* — оптимальное базисное решение задачи (2.13), а y^* — оптимальное решение двойственной задачи. Для простоты будем считать, что x^* — невырожденное базисное решение. Тогда множество $I(x^*)$ ограничений, которые в точке x^* выполняются как равенства, является оптимальным базисным множеством.

Рассмотрим задачу ЛП

$$\max\{c^T x \,:\, Ax \le b + \Delta b,\; x \ge 0\}, \tag{2.14}$$

$\Delta b \in \mathbb{R}^m$. Пусть $\tilde{x}^*$ — оптимальное решение задачи (2.14). Если $\epsilon > 0$ — достаточно малое число и $\|\Delta b\| \le \epsilon$, то $I(\tilde{x}^*) = I(x^*)$ (докажите это). Так как y^* остается допустимым решением двойственной к (2.14) задачи ЛП (ее ограничения не зависят от b), то для пары $(\tilde{x}^*, y^*)$ выполняется условие дополняющей нежесткости (это следует из того, что она выполняется для пары (x^*, y^*)). Теперь мы можем вычислить

$$\frac{\partial z}{\partial b_i}(b) = \lim_{\epsilon \to 0} \frac{z(b + \epsilon \mathrm{e}_i) - z(b)}{\epsilon} = \lim_{\epsilon \to 0} \frac{(b + \epsilon \mathrm{e}_i)^T y^* - b^T y^*}{\epsilon} = y_i^* \,.$$

Экономический смысл двойственных переменных следует из приблизительного равенства

$$z(b + \epsilon \mathrm{e}_i) \approx y_i^* \epsilon,$$

которае означает, что на каждую дополнительную единицу ресурса i предприятие получает прибыль равную y_i^*. Поэтому оптимальные двойственные переменные y_i^* называются *теневыми ценами*. Если теневая цена y_i^* больше цены ресурса i на рынке, то предприятию для увеличения прибыли целесообразно закупить дополнительное количество i-го ресурса. Из условия дополняющей нежесткости $y_i^*(b_i - A_i x^*) = 0$ следует, что теневая цена неполностью использованного ресурса $(A_i x^* < b_i)$ равна нулю.

Приведенной стоимостью переменной x_j (продукта j) называется величина

$$\bar{c}_j = c_j - \sum_{i=1}^{m} a_{ij} y_i^*,$$

равная стоимости единицы продукта j минус теневая стоимость ресурсов, используемых для ее производства. Отметим следующии свойства приведенных стоимостей.

- Поскольку y^* — допустимое решение задачи ЛП

$$\max\{b^T y : \ y^T A \geq c, \ y \geq 0\}$$

двойственной задаче (2.13), то все приведенные стоимости неположительны.

- Из условия дополняющей нежесткости $x_j^* \left(c_j - \sum_{i=1}^{n} a_{ij} y_i^*\right) = 0$ следует, что приведенная стоимость производимого продукта j ($x_j^* > 0$) равна нулю, а если приведенная стоимость отрицательна, то продукт не производится ($\bar{c}_j < 0$ влечет $x_j^* = 0$).

2.2. Симплекс-метод

Решим следующую задачу линейного программирования:

$$\begin{array}{rcrcrcl}
5x_1 & + & 2x_2 & + & 3x_3 & \to & \max, \\
2x_1 & + & 3x_2 & + & x_3 & \leq & 10, \\
4x_1 & + & 2x_2 & + & 2x_3 & \leq & 12, \\
2x_1 & + & x_2 & + & 2x_3 & \leq & 8, \\
 & & & x_1, & x_2, \ x_3 & \geq & 0.
\end{array}$$

Вводя переменные недостатка x_3, x_4 и x_5, запишем эквивалентную задачу:

$$\begin{array}{rcrcrcrcrcrcrcl} 5x_1 & + & 2x_2 & + & 3x_3 & & & & & & & \to & \max, \\ 2x_1 & + & 3x_2 & + & x_3 & + & x_4 & & & & & = & 10, \\ 4x_1 & + & 2x_2 & + & 2x_3 & & & + & x_5 & & & = & 12, \\ 2x_1 & + & x_2 & + & 2x_3 & & & & & + & x_6 & = & 8, \\ & & & & & & \multicolumn{5}{r}{x_1,\ x_2,\ x_3,\ x_4,\ x_5,\ x_6} & \geq & 0. \end{array} \tag{2.15}$$

1. Запишем задачу (2.15) в табличной форме:

	b	x_1	x_2	x_3	x_4	x_5	x_6	Отношения
$-z$	0	5	2	3	0	0	0	
x_4	10	2	3	1	1	0	0	$\frac{10}{2} = 5$
x_5	12	4	2	2	0	1	0	$\frac{12}{4} = 3$
x_6	8	2	1	2	0	0	1	$\frac{8}{2} = 4$

В базис вводим переменную x_1 с максимальным целевым коэффициентом равным 5, при этом, столбец 1 называется *ведущим.* В симплекс-таблице он выделен двойными линиями. Чтобы определить, какая переменная должна покинуть базис, вычисляем отношения элементов столбца b (содержит значения базисных переменных) к положительным элементам ведущего столбца. Эти отношения записаны в последнем столбце симплекс-таблицы. Выбираем в качестве *ведущей* строку с минимальным отношением. В симплек-таблице эта строка выделена двойными линиями. Ей соответствует базисная переменная x_5. Переменная x_5 должна покинуть базис, а ее место должна занять переменная x_1. Итерация симплекс-метода завершается выполнением *операции замещения* с выбранными ведущей строкой и ведущим столбцом. Суть этой операции в том, чтобы с помощью элементарных матричных операций сделать ведущий столбец единичным с единицей в ведущей строке. Пересчитанная симплекс-таблица представлена ниже.

Все последующие итерации выполняются по описанным выше правилам. Мы приводим их без комментариев.

2.

	b	x_1	x_2	x_3	x_4	x_5	x_6	Отно-шения
$-z$	-15	0	$-\frac{1}{2}$	$\frac{1}{2}$	0	$-\frac{5}{4}$	0	
x_4	4	0	2	0	1	$-\frac{1}{2}$	0	—
x_1	3	1	$\frac{1}{2}$	$\frac{1}{2}$	0	$\frac{1}{4}$	0	6
x_6	2	0	0	1	0	$-\frac{1}{2}$	1	2

3.

	b	x_1	x_2	x_3	x_4	x_5	x_6	Отно-шения
$-z$	-16	0	$-\frac{1}{2}$	0	0	-1	$-\frac{1}{2}$	
x_4	4	0	2	0	1	$-\frac{1}{2}$	0	
x_1	2	1	$\frac{1}{2}$	0	0	$\frac{1}{2}$	$-\frac{1}{2}$	
x_3	2	0	0	1	0	$-\frac{1}{2}$	1	

Поскольку все целевые коэффициенты отрицательны, то текущая базисное решение оптимально. Теневые цены с отрицательным знаком записаны в строке $-z$ оптимальной симплек-таблицы в позициях, соответствующих переменным недостатка.

Ответ:
а) оптимальное решение $x^* = (2, 0, 2)$,
б) теневые цены $y^* = (0, 1, 1/2)$.

Решим еще один пример:

$$\begin{array}{rcrcrcl}
2x_1 & + & x_2 & + & x_3 & \to & \max, \\
4x_1 & + & 2x_2 & & & \le & 8, \\
x_1 & + & x_2 & + & x_3 & \le & 5, \\
 & & x_2 & + & 2x_3 & \le & 5, \\
 & & & x_1, & x_2, \ x_3 & \ge & 0.
\end{array}$$

1.

	b	x_1	x_2	x_3	x_4	x_5	x_6	Отно-шения
$-z$	0	2	1	1	0	0	0	
x_4	8	4	2	0	1	0	0	2
x_5	5	1	1	1	0	1	0	5
x_6	5	0	1	2	0	0	1	—

2.

	b	x_1	x_2	x_3	x_4	x_5	x_6	Отношения
$-z$	-4	0	0	1	$-\frac{1}{2}$	0	0	
x_1	2	1	$\frac{1}{2}$	0	$\frac{1}{4}$	0	0	—
x_5	3	0	$\frac{1}{2}$	1	$-\frac{1}{4}$	1	0	3
x_6	5	0	1	2	0	0	1	$\frac{5}{2}$

3.

	b	x_1	x_2	x_3	x_4	x_5	x_6	Отношения
$-z$	$-\frac{13}{2}$	0	$-\frac{1}{2}$	0	$-\frac{1}{2}$	0	$-\frac{1}{2}$	
x_1	2	1	$\frac{1}{2}$	0	$\frac{1}{4}$	0	0	
x_5	$\frac{1}{2}$	0	0	0	$-\frac{1}{4}$	1	$-\frac{1}{2}$	
x_3	$\frac{5}{2}$	0	$\frac{1}{2}$	1	0	0	$\frac{1}{2}$	

Ответ:
a) оптимальное решение $x^* = (2, 0, 5/2)$,
b) теневые цены $y^* = (1/2, 0, 1/2)$.

2.3. Модели линейного программирования

Способность на практике распознать потенциальную задачу ЛП и затем сформулировать ее для решения на компьютере есть в своем роде искусство, которое совершенствуется по мере приобретения практического опыта. В этом разделее мы рассмотрим несколько формулировок задач ЛП различной сложности.

2.3.1. Задача о диете

В *задаче о диете* нужно приготовить блюдо из заданных продуктов (ингредиентов), которое бы удовлетворяло ряду требований. Продемонстрируем это на конкретном при мере.

Диетолог в больнице разрабатывает молочный коктейль для послеоперационных больных. Диетолог хочет, чтобы в коктейле количество холестерола не превышала 175 единиц, а количество насыщенных жиров — 150 единиц. Белков должно быть не меньше 200 единиц, а колорий — не меньше 100 единиц. Диетолог выбрал три возможных ингредиента для коктейля: куриные яйца, мороженое и фруктовый сироп. Информация о стоимости и составе ингредиентов представлена в следующей таблице.

Продукт	Цена	К-во холестерина	К-во жиров	К-во белков	К-во калорий
яйцо	\$0.15	50	0	70	30
мороженое	\$0.25	150	100	10	80
сироп	\$0.10	90	50	0	200

Нужно смешать ингредиенты в таких пропорциях, чтобы удовлетворялись вышеперечисленные требования и стоимость единицы коктейля была минимальной.

Для формулировки данной задачи как задачи ЛП выберем следующие переменные:

- E — количество яиц в единице коктейля;
- C — количество единиц мороженого в единице коктейля;
- S — количество единиц сиропа в единице коктейля.

В этих переменных задача формулируется следующим образом:

$$
\begin{array}{rrrll}
0.15E + & 0.25C + & 0.1S & \to \min, & \\
50E + & 150C + & 90S & \le 175, & \text{(холестерин)} \\
 & 100C + & 50S & \le 150, & \text{(жир)} \\
70E + & 10C & & \ge 200, & \text{(белки)} \\
30E + & 80C + & 200S & \ge 100, & \text{(калории)} \\
 & & E, C, S & \ge 0. &
\end{array}
$$

2.3.2. Арбитраж

В нашем распоряжении имеется n финансовых активов, цена j-го из них в начале инвестиционного периода равна p_j. В конце инвестиционного периода цена актива j есть случайная величина v_j. Предположим, что в конце периода возможны m сценариев (исходов). Тогда v_j есть дискретная случайная величина. Пусть v_{ij} есть значение v_j для сценария i. Из элементов v_{ij} составим $m \times n$-матрицу V.

Торговая политика представляется вектором $x = (x_1, \ldots, x_n)^T$: если $x_j > 0$, то мы покупаем x_j единиц актива j, а если $x_j < 0$, то $-x_j$ единиц актива j продается. Торговая политика называется *арбитражем*, если мы можем заработать сегодня без риска потерь завтра:

$$p^T x < 0, \tag{2.16a}$$

$$Vx \geq 0. \tag{2.16b}$$

Строгое неравенство (2.16a) означает, что в начале периода мы получаем больше, чем тратим. А выполнение всех неравенств $\sum_{j=1}^n v_{ij}x_j \geq 0$ из системы (2.16b) означает, что обратная торговая политика $-x$ не будет убыточной в любом из m возможных сценариев в конце периода.

Поскольку на рынке цены приспосабливаются очень быстро, то возможность заработать на арбитраже тоже исчезает очень быстро. Поэтому в математических финансовых моделях очень часто предполагается, что арбитража не существует.

Теорема 2.2. *Справедливы следующие утверждения.*

а) Арбитраж отсутствует тогда и только тогда, когда совместна следующая система неравенств

$$V^T y = p, \quad y \geq 0.$$

б) При фиксированных ценах $p_1, \dots, p_{j-1}, p_{j+1}, \dots, p_n$, если имеют решения следующие задачи ЛП

$$p_j^{\min} = \min\{p_j : \; V^T y = p, \; y \geq 0, \; p_j \geq 0\},$$
$$p_j^{\max} = \max\{p_j : \; V^T y = p, \; y \geq 0, \; p_j \geq 0\},$$

то арбитраж отсутствует тогда и только тогда, когда

$$p_j^{\min} \leq p_j \leq p_j^{\max}.$$

Доказательство. Решение системы неравенств (2.16) сводится к решению следующей задачи ЛП:

$$\min\{p^T x : \; Vx \geq 0\}, \tag{2.17}$$

причем, арбитраж существует тогда и только тогда, когда целевая функция в задаче (2.17) неограничена. Тогда по теореме двойственности (2.1) двойственная задача ЛП

$$\min\{\mathbf{0}^T y : \; V^T y = p, \; y \geq 0\}$$

не имеет допустимых решений. Обращая это утверждение, получаем утверждение а).

Утверждение б) является простым следствием утверждения а). □

Применим полученные результаты к простой ситуации, когда имеется всего три актива: 1) безрисковый (например, наличность или облигации) с фиксированным возвратом r ($r > 1$) за инвестиционный период, 2) акции XYZ, которые в начале периода продаются по цене S за акцию, и 3) опционы на акции XYZ. Опцион дает право, но не обязывает, по завершении инвестиционного периода, купить акцию XYZ по фиксированной цене K.

Предположим, что в конце инвестиционного периода возможны два сценария. В первом — цена акции XYZ вырастет в u раз ($u > r$), а во втором — цена акции упадет в d раз ($d < 1$).

Все исходные данные можно представить следующим образом:

$$p_1 = 1,\ p_2 = S,\ p_3 = ?; \quad V = \begin{pmatrix} r & uS & \max\{0, uS - K\} \\ r & dS & \max\{0, dS - K\} \end{pmatrix}.$$

При каких стоимостях опциона p_3 отсутствует арбитраж?

2.3.3. Метод DEA

Метод DEA (Data Envelopment Analysis) применяется для сравнения эффективности работы ряда аналогичных сервисных подразделений (отделений банка, ресторанов, учреждений образования, здравоохранения, станций технического обслуживания автомобилей и многих других). Метод DEA не требует стоимостной оценки представляемых услуг. Предположим, что имеется n подразделений, которые занумерованы числами $1, \dots, n$. За тестовый период подразделение i ($i = 1, \dots, n$) использовало r_{ij} единиц ресурса j ($j = 1, \dots, m$) и оказало s_{ik} услуг вида k ($k = 1, \dots, l$). Эффективность работы подразделения i оценивается отношением

$$E_i(u, v) \stackrel{\text{def}}{=} \frac{\sum_{k=1}^{l} s_{ik} u_k}{\sum_{j=1}^{m} r_{ij} v_j},$$

где u_k и v_j есть весовые множители, которые нужно определить.

Чтобы вычислить рейтинг подразделения i_0, нужно решить следующую *задачу дробно-линейного программирования*:

$$\max\{E_{i_0}(u, v) :\ E_i(u, v) \le 1,\ i = 1, \dots, n;\ i \ne i_0;\ u \in \mathbb{R}^l_+,\ v \in \mathbb{R}^m_+\}.$$

Эту задачу можно переформулировать как следующую задачу ЛП:

$$\sum_{k=1}^{l} s_{i_0 k} u_k \to \max, \tag{2.18a}$$

$$\sum_{j=1}^{m} r_{i_0 j} v_j = 1, \tag{2.18b}$$

$$\sum_{k=1}^{l} s_{ik} u_k \le \sum_{j=1}^{m} r_{ij} v_j, \quad i = 1, \dots, n;\ i \ne i_0 \tag{2.18c}$$

$$u_k \ge 0, \quad k = 1, \dots, l, \tag{2.18d}$$

$$v_j \ge 0, \quad j = 1, \dots, m. \tag{2.18e}$$

Пусть (u^*, v^*) есть оптимальное решение задачи (2.18). Если $E_{i_0}(u^*, v^*) < 1$, то подразделение i_0 работало неэффективно, и его работу можно улучшить, если перенять опытом работы у более эффективных подразделений i, для которых $E_i(u^*, v^*) = 1$.

Демонстрационный пример

Фирма быстрого питания имеет шесть подразделений, каждое из которых размещено в одном из торговых центров с большой парковкой. Фирма предлагает клиентам только один стандартный набор, включающий бургергер, картофель фри и напиток. Мэнеджеры фирмы решили использовать DEA, чтобы выявить те подраздения, которые используют свои ресурсы наиболее эффективно. Данные для DEA анализа представлены в таблице 2.3.3.

Чтобы вычислить рейтинг подразделения 1, мы формулируем следующую задачу ЛП:

$$\begin{aligned} E_1 = 1600u_1 &\to \max, \\ 32v_1 - 3200v_2 &= 1, \\ 400u_1 - 16v_1 - 600v_2 &\le 0, \\ 600u_1 - 24v_1 - 600v_2 &\le 0, \\ 400u_1 - 24v_1 - 400v_2 &\le 0, \\ 200u_1 - 16v_1 - 160v_2 &\le 0, \\ 80u_1 - 8v_1 - 40v_2 &\le 0, \\ u_1, v_1, v_2 &\ge 0. \end{aligned}$$

Таблица 2.2

Данные для DEA анализа

Подраз-деление	Труд (часов)	Материалы (долларов)	Наборов продано
1	32	3200	1600
2	16	600	400
3	24	600	600
4	24	400	400
5	16	160	200
6	8	40	80

2.3.4. Краткосрочный финансовый менеджмент

Финансовый менеджмент в краткосрочной перспективе есть одна из задач бухгалтерии большой фирмы. При неудачном управлении финансами доходы получат банки, в которых хранятся денежные средства, а не их владелец. Свободные деньги также должны работать. Прибыль можно существенно увеличить, если работать активно на рынке ценных бумаг.

Предположим, что плановый горизонт разделен на T периодов различной продолжительности; период $T+1$ представляет конец горизонта. На рынке имеется n типов ценных бумаг. Портфель фирмы в начале планового горизонта представлен вектором s размера n, где $s_i \geq 0$ есть число ценных бумаг типа i в портфеле. Стоимости продажи и покупки одной ценной бумаги типа i в период t обозначаются через c^s_{it} и c^b_{it}, соответственно. Заметим, что величины c^s_{it} и c^b_{it} могут быть меньше или больше, чем номинальная стоимость ценной бумаги типа i.

Краткосрочные финансовые источники (отличные от продажи бумаг из портфеля) представлены k открытыми кредитными линиями. Максимальный объем заимствований по линии l обозначим через u_l. Заемы можно получить в начале каждого периода, а возвращать нужно уже после завершения планового горизонта. Чтобы оценить эффективность использования заемных средств, вычислены издержки f_{lt} при получении единицы заема по линии l в период t: величина f_{lt} равна месячному проценту, умноженному на время (в месяцах), оставшееся до конца планового горизонта.

Экзогенные (внешние) денежные потоки заданы величинами d_t, $t = 1, \dots, T$. Если $d_t > 0$ (соответственно $d_t < 0$), то фирма должна получить сумму d_t (заплатить $-d_t$) в начале периода t. Считаем, что запас наличности в начале планового горизонта учтен при вычислении d_1.

Для каждого периода $t = 1, \dots, T$ задана также минимальная потребность в наличности q_t.

Нужно сбалансировать бюджет наличности таким образом, чтобы максимизировать «богатство» (наличность плюс продажная стоимость всех ценных бумаг минус сумма всех заемов с учетом процентов) фирмы в конце планового горизонта.

Определим следующие переменные:

- x_{it} — число ценных бумаг типа i в конце периода t;
- x_{it}^s — число ценных бумаг типа i, проданных в период t;
- x_{it}^b — число ценных бумаг типа i, купленных в период t;
- y_t — наличность в период t;
- z_{lt} — заем, полученный по кредитной линии l в период t.

Теперь мы можем записать следующую задачу СЦП:

$$y_T + \sum_{i=1}^{n} c_{i,T+1}^s x_{i,T} - \sum_{t=1}^{T}\sum_{l=1}^{k}(1 + f_{lt}) z_{lt} \to \max, \tag{2.19a}$$

$$d_1 + \sum_{i=1}^{n} c_{i1}^s x_{i1}^s + \sum_{l=1}^{k} z_{l1} = y_1 + \sum_{i=1}^{n} c_{i1}^b x_{i1}^b, \tag{2.19b}$$

$$d_t + y_{t-1} + \sum_{i=1}^{n} c_{it}^s x_{it}^s + \sum_{l=1}^{k} z_{lt} = y_t + \sum_{i=1}^{n} c_{it}^b x_{it}^b, \quad t = 2, \dots, T, \tag{2.19c}$$

$$s_i + x_{i1}^b - x_{i1}^s = x_{i1}, \quad i = 1, \dots, n, \tag{2.19d}$$

$$x_{i,t-1} + x_{it}^b - x_{it}^s = x_{it}, \quad i = 1, \dots, n;\ t = 2, \dots, T, \tag{2.19e}$$

$$\sum_{t=1}^{T} z_{lt} \le u_l, \quad l = 1, \dots, k, \tag{2.19f}$$

$$y_t \ge q_t, \quad t = 1, \dots, T, \tag{2.19g}$$

$$x_{it},\ x_{it}^s,\ x_{it}^b \in \mathbb{Z}, \quad i = 1, \dots, n;\ t = 1, \dots, T, \tag{2.19h}$$

$$y_t \in \mathbb{R}_+, \quad t = 1, \dots, T, \tag{2.19i}$$

$$z_{lt} \in \mathbb{R}_+, \quad l = 1, \dots, k;\ t = 1, \dots, T. \tag{2.19j}$$

Целевая функция (2.19a) требует максимизировать «богатство» фирмы в конце планового горизонта. Ограничения (2.19c) и (2.19e) балансируют бюджеты соответственно наличности и ценных бумаг в периоды $2, \dots, T$. Аналогичные балансовые ограничения (2.19b) и (2.19d) применимы только для периода 1. Неравенства (2.19f) гарантируют, что суммарный заем по любой кредитной линии не должен превышать объем этой линии. Неравенства (2.19g) требуют, чтобы в любой период имелся в наличии требуемый минимум денег.

2.3.5. Предсказание предпочтений потребителя

Корзина товаров представляется вектором $x \in [0,1]^n$, где x_i есть количество продукта i в корзине. Предполагается, что все объемы продуктов нормализованы так, что максимальное количество каждого продукта в корзине равно 1. Когда заданы две корзины продуктов x^1 и x^2, потребитель может предпочесть корзину x^1 корзине x^2, предпочесть корзину x^2 корзине x^1, или рассматривать обе корзины x^1 и x^2 равноценными.

Будем считать, что предпочтения потребителя описываются некоторой неизвестной функцией полезности $u : [0,1]^n \to \mathbb{R}$, где $u(x)$ есть мера полезности, извлекаемой потребителем из корзины x. Из двух корзин потребитель выбирает корзину с большим значением полезности, а если для обеих корзин значения функции полезности одинаковы, то для потребителя такие корзины равноценны. Будем стандартно предполагать, что u неубывающая (потребитель предпочитает иметь большее количество любого из продуктов) вогнутая (это свойство моделирует *насыщение* и известно как *закон убывания предельной полезности* при увеличении количества продукта) функция.

Теперь предположим, что потребителя интересуют только конечный набор корзин $x^1, \dots, x^m$, и мы уже знаем ряд предпочтений потребителя. Эти предпочтения заданы

1) множеством пар $\mathcal{P} \in \{1, \dots, m\}^2$ *строгих предпочтений*: для $(i,j) \in \mathcal{P}$ потребитель предпочитает корзину x^i корзине x^j;
2) множеством пар $\mathcal{P}_{\text{week}} \in \{1, \dots, m\}^2$ *нестрогих предпочтений*: для $(i,j) \in \mathcal{P}_{\text{week}}$ потребитель считает, что корзина x^i не хуже корзины x^j.

В терминах функции полезности u предпочтения потребителя задаются неравенствами:

$$
\begin{aligned}
&u(x^i) > u(x^j), \quad (i,j) \in \mathcal{P},\\
&u(x^i) \geq u(x^j), \quad (i,j) \in \mathcal{P}_{\text{week}}.
\end{aligned} \tag{2.20}
$$

Поскольку функции u и tu ($t > 0$) задают одни и те же предпочтения потребителя, то мы можем переформулировать (2.20) в следующем виде:

$$
\begin{aligned}
&u(x^i) \geq u(x^j) + 1, \quad (i,j) \in \mathcal{P},\\
&u(x^i) \geq u(x^j), \quad (i,j) \in \mathcal{P}_{\text{week}}.
\end{aligned} \tag{2.21}
$$

Пусть $u_i \stackrel{\text{def}}{=} u(x^i)$ и пусть g^i обозначает субградиент функции u в точке x^i. В силу (2.21) и теоремы A.5 ($-u$ — выпуклая функция) должны выпоняться неравенства

$$u_i - u_j \geq 1, \quad (i,j) \in \mathcal{P}, \tag{2.22a}$$

$$u_i - u_j \geq 0, \quad (i,j) \in \mathcal{P}_{\text{week}}, \tag{2.22b}$$

$$u_i - u_j \geq (g^i)^T(x^i - x^j), \quad i,j = 1,\dots,m, \tag{2.22c}$$

$$g^i_1,\dots,g^i_n \geq 0, \quad i = 1,\dots,m. \tag{2.22d}$$

Здесь неравенства (2.22c) выражают тот факт, что функци u вогнутая. Неотрицательность субградиентов, выраженная неравенствами (2.22d), равносильна требованию, чтобы функция u была неубывающей.

Решая систему (2.22), мы можем определить, имеется ли хоть одна функция полезности совместимая с заданным множеством $\mathcal{P}$ строгих и множеством $\mathcal{P}_{\text{week}}$ нестрогих предпочтений потребителя. Если система (2.22) имеет решение $u_1,\dots,u_m; g^1,\dots,g^m$, то по теореме 3.1 мы може определить

$$u(x) \stackrel{\text{def}}{=} \min_{1 \leq i \leq m} \left(u_i + (g^i)^T(x - x^i)\right).$$

Если система (2.22) не имеет решения, то мы можем заключить, что не существует неубываюшей вогнутой функции полезностей совместимой с множествами предпочтений $\mathcal{P}$ и $\mathcal{P}_{\text{week}}$.

Теперь предположим, что система (2.22) совместна. Рассмотрим пару $(k,l) \notin \mathcal{P} \cup \mathcal{P}_{\text{week}}$. Это означает, что предпочтение потребителя для пары корзин (x^k, x^l) неизвестно. В некоторых случаях, мы можем узнать это предпочтение не спрашивая потребителя. Действительно, добавим неравенство $u_k \geq u_l$ к системе (2.22), если в результате получится несовместная система, то мы можем заключить, что потребитель предпочитает корзину x^l корзине x^k. Аналогично, если после добавления к системе

(2.22) неравенства $u_l \geq u_k$ получится несовместная система неравенств, то мы можем заключить, что потребитель предпочитает корзину x^k корзине x^l.

2.3.6. Проверка гипотез

Пусть X есть дискретная случайная величина со значениями из множества $\{1, \ldots, n\}$ и распределением вероятностей, которое зависит от параметра $\theta \in \{1, \ldots, m\}$. Распределения вероятностей случайной величины X для m возможных значений параметра θ представлены $n \times m$-матрицей P с элементами $p_{kj} = \mathbb{P}(X = k|\theta = i)$. Заметим, что i-й столбец P задает распределение вероятностей для X при условии, что $\theta = i$.

Рассмотрим задачу оценки параметра θ по наблюдению (выборке) случайной величины X. Иными словами, значение случайной величины X генерируется для одного из m возможных распределений (значений параметра θ), и мы хотим определить это распределение (значение параметра). Значения параметра θ называют *гипотезами*, а угадывание, какая из m гипотез верна, называют *проверкой гипотез*.

Вероятностный классификатор для θ — это дискретная случайная величина $\hat{\theta}$, которая зависит от наблюдаемого значения X и принимает значения $1, \ldots, m$. Вероятностный классификатор можно представить $m \times n$-матрицей T с элементами $t_{ik} = \mathbb{P}(\hat{\theta} = i|X = k)$. Если мы наблюдаем значение $X = k$, то классификатор в качестве оценки параметра θ выбирает значение $\hat{\theta} = i$ с вероятностью t_{ik}. Качество классификатора можно определить по $m \times m$-матрице $D = TP$ с элементами $d_{ij} = \mathbb{P}(\hat{\theta} = i|\theta = j)$, т. е. d_{ij} есть вероятность предсказания $\hat{\theta} = i$, когда $\theta = j$.

Нужно определить вероятностный классификатор, для которого максимальная из вероятностей ошибок классификации

$$1 - d_{ii} = \sum_{j \neq i} d_{ij} = \mathbb{P}(\hat{\theta} \neq i|\theta = i), \quad i = 1, \ldots, m,$$

минимальна. Данная задача формулируется как задача ЛП следующим образом:

$$\sum_{i=1}^{m} d_{ii} \to \min, \tag{2.23a}$$

$$\sum_{k=1}^{n} p_{kj} t_{ik} - d_{ij} = 0, \quad i,j = 1,\dots,m, \tag{2.23b}$$

$$\sum_{i=1}^{m} t_{ik} = 1, \quad k = 1,\dots,n, \tag{2.23c}$$

$$t_{ik} \ge 0, \quad i,\dots,m;\ k = 1,\dots,n. \tag{2.23d}$$

Здесь равенства (2.23b) выражают матричное равенство $D = TP$. Равенства (2.23c) и неравенства (2.23d) требуют, чтобы переменные t_{ik} принимали значения, соответствующие определению: $t_{ik} = \mathbb{P}(\hat{\theta} = i | X = k)$.

2.4. Транспортная задача

Транспортная задача — это один из самых знаменитых частных случаев задачи ЛП. Имеется m поставщиков и n потребителей некоторого продукта. Поставщик i имеет в наличии a_i единиц данного продукта, а потребитель j хочет получить b_j единиц продукта. Стоимость транспортировки единицы продукта от поставщика i потребителю j равна c_{ij}. Пусть x_{ij} обозначает количество продукта, доставляемого поставщиком i потребителю j. Нужно определить план поставок $X = [x_{ij}]$, для которого суммарные транспортные расходы минимальны.

ЛП формулировка транспортной задачи записывается следующим образом:

$$\sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij} \to \min, \tag{2.24a}$$

$$\sum_{j=1}^{n} x_{ij} \le a_i, \quad i = 1,\dots,m, \tag{2.24b}$$

$$\sum_{i=1}^{m} x_{ij} = b_j, \quad j = 1,\dots,n, \tag{2.24c}$$

$$x_{ij} \ge 0, \quad i = 1,\dots,m;\ j = 1,\dots,n. \tag{2.24d}$$

Здесь целью (2.24a) является минимизация суммарных транспортных издержек. Неравенства (2.24b) требуют, чтобы суммарный объем поставок каждого поставщика не превосходил его возможностей. Равенства (2.24c) гарантируют, что каждый потребитель получит столько, сколько ему нужно.

2.4.1. Метод потенциалов

Метод потенциалов — это симплекс-метод, примененный к задаче (2.24) в предположении, что предложение равно спросу:

$$exc = \sum_{i=1}^{m} a_i - \sum_{j=1}^{n} b_j = 0. \tag{2.25}$$

Если это не так, то

- если $exc < 0$, нужно ввести фиктивного поставщика с предложением $a_{m+1} = -exc$ и стоимостями поставок $c_{m+1,j} = 0$, $j = 1, \ldots, n$;
- если $exc > 0$, нужно ввести фиктивного потребителя со спросом $b_{n+1} = exc$ и стоимостями поставок $c_{i,n+1} = 0$, $i = 1, \ldots, m$.

При выполнении условия (2.25), неравенства (2.24b) можно записать как равенства. Запишем двойственную к такой модифицированной задаче (2.24):

$$\sum_{i=1}^{m} a_i \alpha_i + \sum_{j=1}^{n} b_j \beta_j \to \max, \tag{2.26a}$$

$$\alpha_i + \beta_j \le c_{ij}, \quad i = 1, \ldots, m;\ j = 1, \ldots, n. \tag{2.26b}$$

В методе потенциалов числа α_i и β_j называются *потенциалами*. Заметим, что для допустимого решения (α, β) двойственной задачи (2.26) все приведенные стоимости $\bar{c}_{ij} \stackrel{\text{def}}{=} c_{ij} - \alpha_i - \beta_j$ неотрицательны. Из теоремы двойственности ЛП вытекает следующий критерий оптимальности.

Теорема 2.3. *Допустимое решение x задачи (2.24) является оптимальным тогда и только тогда, когда существуют такие потенциалы (α, β), что*

$$\bar{c}_{ij} \ge 0, \quad i = 1, \ldots, m;\ j = 1, \ldots, n, \tag{2.27}$$

$$\bar{c}_{ij} x_{ij} = 0, \quad i = 1, \ldots, m;\ j = 1, \ldots, n. \tag{2.28}$$

Метод потенциалов на каждой итерации для текущего решения x вычисляет потенциалы (α, β), которые удовлетворяют условию дополняющей нежесткости (2.28). Если при этом окажется, что все приведенные стоимости неотрицательны, то по теореме 2.3 решение x оптимально.

Транспортная таблица

Транспортная таблица имеет следующий вид:

c_{11} $\bar{c}_{11}$ x_{11}	c_{12} $\bar{c}_{12}$ x_{12}	$\cdots$	c_{1j} $\bar{c}_{1j}$ x_{1j}	$\cdots$	c_{1n} $\bar{c}_{1n}$ x_{1n}	α_1
c_{21} $\bar{c}_{21}$ x_{21}	c_{22} $\bar{c}_{22}$ x_{22}	$\cdots$	c_{2j} $\bar{c}_{2j}$ x_{2j}	$\cdots$	c_{2n} $\bar{c}_{2n}$ x_{2n}	α_2
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
c_{i1} $\bar{c}_{i1}$ x_{i1}	c_{i2} $\bar{c}_{i2}$ x_{i2}	$\cdots$	c_{ij} $\bar{c}_{ij}$ x_{ij}	$\cdots$	c_{in} $\bar{c}_{in}$ x_{in}	α_i
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
c_{m1} $\bar{c}_{m1}$ x_{m1}	c_{m2} $\bar{c}_{m2}$ x_{m2}	$\cdots$	c_{mj} $\bar{c}_{mj}$ x_{mj}	$\cdots$	c_{mn} $\bar{c}_{mn}$ x_{mn}	α_m
β_1	β_2	$\cdots$	β_j	$\cdots$	β_n	

Клетки таблицы, которые соответствуют базисным переменным, называются *базисными*. Поскольку ранг матрицы ограничений задачи (2.25) равен $m + n - 1$, то базисных клеток тоже должно быть $m + n - 1$. Так как значения всех небазисных переменных равно нулю, то при ручном счете значения $x_{ij} = 0$ в небазисных клетках не проставляются. Это также позволяет отличать базисные клетки от небазисных. Кроме этого, поскольку приведенные стоимости всех базисных переменных равны нулю, то значения $\bar{c}_{ij} = 0$ также не записываются в таблицу.

Построение начального плана поставок: метод северо-западного угла

Существует несколько способов построить начальный допустимый план перевозок (допустимое решение задачи (2.24)) x. Здесь мы рассмотрим простейший из них: *метод северо-западного угла.*

На каждом шаге метод «вычеркивает» один столбец или одну строку транспортной таблицы. Пусть $\bar{a} = a$, $\bar{b} = b$, и $x = 0$. На шаге

$k = 1, \ldots, m+n-1$ выбирается клетка (i,j) в левом верхнем (северо-западном) углу невычеркнутой части таблицы и полагается

$$x_{ij} = \min\{\bar{a}_i, \bar{b}_j\}, \quad \bar{a}_i := \bar{a}_i - x_{ij}, \quad \bar{b}_j := \bar{b}_j - x_{ij}.$$

Если $\bar{a}_i = 0$, то из таблицы «вычеркивается» i-я строка; в противном случае вычеркивается j-й столбец.

Вычисление потенциалов и приведенных стоимостей

Одному (любому) из потенциалов можно присвоить произвольное значение. Например, $\alpha_1 = 0$. Если еще не все потенциалы вычислены, находим базисную клетку (i,j), для которой вычислен ровно один потенциал, α_i или β_j. Другой потенциал вычисляем из равенства $\alpha_i + \beta_j = c_{ij}$.

Когда все потенциалы α_i и β_j определены, для каждой небазисной клетки (i,j) вычисляем приведенную стоимость по правилу:

$$\bar{c}_{ij} = c_{ij} - \alpha_i - \beta_j\,.$$

Построение нового плана поставок

Находим клетку (k,l) с минимальной приведенной стоимостью $\bar{c}_{kl}$. Если $\bar{c}_{kl} \geq 0$, то текущий план поставок оптимален и метод заканчивает работу. В противном случае находим единственный цикл

$$(k,l) = (i_0, j_0), (i_0, j_1), (i_1, j_1), \ldots, (i_{s-1}, j_s), (i_s, j_s) = (i_0, j_0),$$

который начинается и заканчивается в клетке (k,l), а все его промежуточные клетки

$$(i_0, j_1), (i_1, j_1), \ldots, (i_{s-1}, j_s)$$

являются базисными. Неформально, из клетки $(i_0, j_0) = (k,l)$ мы идем по строке i_0 до базисной клетки (i_0, j_1), в которой цикл делает изгиб и мы идем уже по столбцу до базисной клетки (i_1, j_1). Далее по столбцу идем до клетки (i_1, j_2), и так далее, пока не вернемся в начальную клетку $(i_s, j_s) = (k,l)$. Мы здесь не описываем формальную процедуру для нахождения такого цикла. Это будет сделано позже в разделе 6.4, где мы изучаем более общий вариант транспортной закдачи — сетевую транспортную задачу.

Затем вычисляем минимальное значение поставки в нечетных клетках цикла

$$\epsilon = \min\{x_{i_0,j_1}, x_{i_1,j_2}, \ldots, x_{i_{s-1},j_s}\}$$

и уменьшаем на ϵ поставки в нечетных клетках и увеличиваем на ϵ поставки в четных клетках:

$$x_{i_{t-1},j_t} := x_{i_{t-1},j_t} - \epsilon, \quad x_{i_t,j_t} := x_{i_t,j_t} + \epsilon, \quad t = 1,\ldots,s.$$

2.4.2. Численный пример

Фирма, представляющая в аренду автомобили по всей стране, обнаружила дисбаланс в распределении автомобилей. В городах 1,2 и 3 имеется избыточное количество автомобилей: 26 — в городе 1, 43 — в городе 2, 31 — в городе 3. В городах 4,5,6 автомобилей не хватает: 32 — в городе 4, 28 — в городе 5, 26 — в городе 6. Расстояния между городами следующие

	4	5	6
1	120	70	350
2	156	240	75
3	225	160	145

Затраты по перегону автомобиля из одного города в другой пропорциональны расстоянию между городами. Разработайте самый экономный план передислокации автомобилей.

Это транспортная задача, в которой поставщиками (автомобилей) являются города 1,2 и 3, а потребителями — города 5,6 и 7. Решим задачу методом потенциалов.

Поскольку суммарное предложение $32+28+26 = 100$ больше спроса $26+43+31 = 86$, то вводим фиктивного потребителя со спросом 14. Начальный план строим по методу северо-западного угла. Ниже приведены таблицы на всех итерациях метода потенциалов.

Итерация 1.

	32	28	26	14	
26	120 **26**	70 -134	350 311	0 106	0
43	156 **6**	240 **28** -	75 **9** +	0 70	36
31	225 -1	160 -150 +	145 **17** -	0 **14**	106
	120	204	39	-106	

Выбираем клетку $(3,2)$ с минимальной приведенной стоимостью и ищем цикл, содержащий данную клетку. Минимальное значение поставки в клетках, отмеченных знаком «-», равно $\epsilon = \min\{17, 28\} = 17$. Увеличиваем значения поставок в клетках, отмеченных знаком «+», и уменьшаем значения поставок в клетках, отмеченных знаком «-». Результат представлен в таблице, с которой начинается итерация 2.

Итерация 2.

120		70	-134	350	311	0	-44	0
26	-		+					
156		240		75		0	-80	36
6	+	**11**	-	**26**				
225	149	160		145	150	0		-44
		17				14		
120		204		39		44		

Выбираем клетку $(1,3)$ с минимальной приведенной стоимостью и ищем цикл, содержащий данную клетку. Минимальное значение поставки в клетках, отмеченных знаком «-», равно $\epsilon = \min\{26, 11\} = 11$. Увеличиваем значения поставок в клетках, отмеченных знаком «+», и уменьшаем значения поставок в клетках, отмеченных знаком «-». Результат представлен в таблице, с которой начинается итерация 3.

3.

120		70		350	311	0	90	0
15		**11**						
156		240	134	75		0	54	36
17				**26**				
225	15	160		145	16	0		90
		17				14		
120		70		39		-90		

Поскольку приведенные стоимости во всех клетках неотрицательны, то текущий план оптимален.

Ответ: перегнать

- из города 1: 15 автомобилей в город 4 и 11 автомобилей в город 5;

- из города 2: 17 автомобилей в город 4 и 26 автомобилей в город 6;
- из города 3: 17 автомобилей в город 5.

2.4.3. Агрегированное планирование

Предприятие, которое осуществляет сборку автомобилей, должно разработать агрегированный план на месяц, разделенный на четыре недели. Прогнозируется спрос на 100 автомобилей в каждую неделю. Производственные возможности предприятия составляют 440 автомобилей:

- недели 1,4: 60 в регулярное время и 20 во внеурочное.
- недели 2,3: 100 в регулярное время и 40 во внеурочное.

Поскольку проиводственные возможности превышают спрос на 40 автомобилей $(2 \cdot (60 + 20 + 100 + 40) - 4 \cdot 100)$, то решено к концу месяца накопить на складе 20 автомобилей. В начале месяца на складе имеется 10 автомобилей.

Затраты на сборку одного автомобиля в регулярное время — \$150, а во внеурочное — \$200. Стоимость хранения одного автомобиля в течении недели равна \$5. Допускаются поставки автомобилей с задержкой в одну неделю. При этом дилеры получают автомобили со скидкой \$50.

Нужно определить сколько автомобилей производить в каждую из недель, чтобы полностью удовлетворить спрос с минимальными затратами (на сборку и хранение автомобилей + скидки за поздние поставки).

Данная задача формулируется как транспортная задача, представленная в табл. 2.3. Начальный план построен методом северо-западного угла. Найдите оптимальный план.

2.5. Упражнения

2.1. Фирма упаковывает и продает в пачках по 250 грамм три марки молотого кофе: 1) изысканный, 2) ароматный и 3) крепкий соответственно по цене \$2.60, \$2.50 и \$2.30 за пачку. Каждая из марок кофе — это смесь двух сортов кофейных зерен: колумбийских и бразильских. Процент колумбийского кофе в марке 1 — 80 , % марке 2 — 50 , % марке 3 — 30 . % Фирма имеет 200 кг. колумбийских зерен, купленных по цене \$3.60 за кг., и 300 кг. бразильских зерен, купленных по цене \$2 за кг. Прожарка, перемолка и упаковка пачки кофе стоит \$0.30.

Фирма должна решить сколько пачек каждой из трех марок кофе

Таблица 2.3

Агрегированное планирование методом потенциалов

Периоды производства		Периоды потребления 1	2	3	4	Склад	Неисп. мощн.	Произв. мощн.
Склад		0 10	5	10	15	20	0	10
1	Регулярное время	150 60	155	160	165	170	0	60
	Внеурочное время	200 20	205	210	215	220	0	20
2	Регулярное время	200 10	150 90	155	160	165	0	100
	Внеурочное время	250	200 10	205 30	210	215	0	40
3	Регулярное время	∞	200	150 70	155 30	160	0	100
	Внеурочное время	∞	250	200	205 40	210	0	40
4	Регулярное время	∞	∞	200	150 30	155 20	0 10	60
	Внеурочное время	∞	∞	250	200	205	0 20	20
Спрос		100	100	100	100	20	30	450

произвести, чтобы получить наибольшую чистую прибыль.

2.2. Портфельный менеджер банка должен определить структуру портфеля из 5 акций, которые описаны в следующей таблице:

Акция	Эмитент	Рейтинг		Лет до погашения	Доходность (% в год)	Налог (%)
		Moody's	Банка			
M_1	Муниципалитет	Aa	2	9	4.3	0
M_2	Муниципалитет	Ba	5	2	4.5	0
G_1	Правительство	Aaa	1	4	5.0	50
G_2	Правительство	Aaa	1	3	4.4	50
A	Агенство	Aaa	1	3	4.4	50

Инвестиционная политика банка накладывает следующие ограничения при формировании портфеля:

1) доля муниципальных облигаций не должна превосходить 60 %;
2) среднее качество акций в портфеле не должно быть большим 1.4 по банковской шкале (заметим, что чем меньше банковский рейтинг, тем выше качество акции);
3) среднее время до погашения, вычисленное по всем акциям в портфеле с учетом их доли, не должно превышать пяти лет.

Цель менеджера — составить портфель $x = (x_{M_1}, x_{M_2}, x_{G_1}, x_{G_1}, x_A)$ с максимальной годовой доходностью (на каждый инвестированный доллар). Здесь $x_{M_1}, x_{M_2}, x_{G_1}, x_{G_1}, x_A$ обозначают доли средств, вложенных соответственно в облигации M_1, M_2, G_1, G_1, A.

а) Сформулируйте эту задачу как задачу ЛП.

б) В предположении, что решено сформировать портфель только из акций M_1, G_2 и A, запишите задачу ЛП и найдите оптимальный портфель графическим методом.

2.3. Используя правила из табл. 2.1, запишите двойственные для следующих задач ЛП:

а)
$$\begin{aligned} 2x_1 - 4x_2 + 3x_3 &\to \max, \\ x_1 + x_2 - x_3 &= 9, \\ -2x_1 + x_2 &\le 5, \\ x_1 - 3x_3 &\ge 4, \\ x_1 &\ge 0, \\ x_3 &\le 0. \end{aligned}$$

б)
$$\begin{aligned} 5x_1 - x_2 + 4x_3 &\to \max, \\ x_1 + x_2 + x_3 &= 12, \\ 3x_1 - 2x_3 &\ge 1, \\ x_2 - x_3 &\le 2, \\ x_1, x_3 &\ge 0; \end{aligned}$$

2.4. Решите следующие задачи ЛП:

а)
$$\begin{aligned} &5x_1+3x_2+4x_3\to\max,\\ &2x_1+\ x_2+\ x_3\le 4,\\ &x_1+\ x_2+2x_3\le 3,\\ &x_1+3x_2+\ x_3\le 9,\\ &x_1,x_2,x_3\ge 0, \end{aligned}$$

б)
$$\begin{aligned} &3x_1+x_2+4x_3\to\max,\\ &x_1+x_2+\ x_3=8,\\ &3x_1\qquad -2x_3\ge 1,\\ &x_2-\ x_3\le 2,\\ &x_1,x_2,x_3\ge 0. \end{aligned}$$

2.5. Найти оптимальное решение задачи ЛП

$$\begin{aligned} &5x_1+4x_2+\ x_3\to\max,\\ &4x_1+2x_2-\ x_3\le 8,\\ &3x_1-\ x_2+2x_3\le 5,\\ &x_1+2x_2+2x_3\le 6,\\ &x_1,x_2,x_3\ge 0, \end{aligned}$$

если известно, что вектор теневых цен следующий: $y^*=(1,0,1)^T$.

2.6. Рассмотрим задачу ЛП

$$\max\left\{\sum_{j=1}^n c_jx_j\ :\ \sum_{j=1}^n a_jx_j\le b,\ 0\le x_j\le u_j,\ j=1,\dots,n\right\} \tag{2.29}$$

с $c_j,a_j>0$ для всех $j=1,\dots,n$. Пусть перестановка $\pi:\{1,\dots,n\}\to\{1,\dots,n\}$ такова, что

$$\frac{c_{\pi(1)}}{a_{\pi(1)}}\ge\frac{c_{\pi(2)}}{a_{\pi(2)}}\ge\dots\ge\frac{c_{\pi(n)}}{a_{\pi(n)}}\,,$$

и

$$\sum_{j=1}^{r-1}a_{\pi(j)}\,u_{\pi(j)}\le b,\quad \text{а}\quad \sum_{j=1}^{r}a_{\pi(j)}\,u_{\pi(j)}>b.$$

Покажите, что компоненты оптимального решения задачи (2.29) определяются по правилу:

$$\begin{aligned} &x^*_{\pi(j)}=u_{\pi(j)},\quad j=1,\dots,r-1,\\ &x^*_{\pi(r)}=\frac{b-\sum_{j=1}^{r-1}a_{\pi(j)}\,u_{\pi(j)}}{a_{\pi(r)}},\\ &x^*_{\pi(j)}=0,\quad j=r+1,\dots,n. \end{aligned}$$

2.7. *Переопределенные системы линейных уравнений.* Задана $m \times n$-матрица A и вектор $b \in \mathbb{R}^m$. Если $m > n$, то система $Ax = b$ может не иметь решения. В таких случаях в качестве решения системы ищут такой вектор x, для которого вектор невязок $Ax - b$ имеет минимальную норму. На практике чаще всего применяются нормы l_2, l_1 и l_∞. В зависимости от типа нормы нужно решать одну из следующих задач безусловной оптимизации:

$$\|Ax - b\|^2 = \sum_{i=1}^{m} (A_i x - b_i)^2 \to \min, \tag{2.30}$$

$$\|Ax - b\|_1 = \sum_{i=1}^{m} |A_i x - b_i| \to \min, \tag{2.31}$$

$$\|Ax - b\|_\infty = \max_{1 \le i \le m} |A_i x - b_i| \to \min. \tag{2.32}$$

Задача (2.30) решается просто: ее решениями являются решения системы уравнений $A^T A x = A^T b$ (докажите это!). Если A есть матрица полного столбцового ранга ($\mathrm{rank} A = n$), то задача (2.30) имеет единственное аналитическое решение $x = \left(A^T A\right)^{-1} A^T b$ (сравните это решение с решением задачи регуляризации Тихонова, сформулированной в упр. 3.1 при $\delta = 0$).

Две другие задачи, которые не являются задачами гладкой оптимизации, решить труднее. Переформулируйте задачи (2.31) и (2.32) как задачи ЛП.

2.8. Покажите, что каждая задача ЛП может быть сведена к задаче

$$\begin{aligned}
&\lambda \to \max,\\
&\sum_{i=1}^{m} t_{ij} = 1, \quad i = 1, \dots, n,\\
&\sum_{j=1}^{m} \sum_{i=1}^{n} a_{ijk} t_{ij} = \lambda, \quad k = 1 \dots, q,\\
&t_{ij} \ge 0, \quad i = 1, \dots, m;\ j = 1, \dots, n,
\end{aligned}$$

которую исследовал Л. В. Канторович. Эта задача допускает следующую интерпретацию. Для производства единицы конечного продукта требуется по единице каждого из q промежуточных продуктов. Имеется n станков, которые могут выполнять m заданий. При выполнении станком i задания j за смену производится a_{ijk} единиц промежуточного продукта k ($k = 1, \ldots, q$). Если t_{ij} есть доля времени работы станка i над заданием j, то λ — это количество произведенных единиц конечного продукта.

2.9. Фирма имеет три предприятия, которые производят одну и ту же продукцию. Производственные издержки и стоимость сырья (на единицу продукта) для всех предприятий различны. Имеется четыре оптовых склада, где потребители покупают продукцию фирмы, причем цена на нее на каждом складе разная. Найти оптимальный план производства и распределения продукции по складам для исходных данных, представленных в следующей таблице.

Предприятие		1	2	3		
Издержки произ-ва		15	20	13		
Стоимость сырья		10	9	12		
		Транспортные изд-ки			Спрос	Цена
Склады	1	3	9	5	80	34
	2	1	7	4	110	32
	3	5	8	3	150	31
	4	7	3	8	100	30
Произв. мощности		140	180	150		

2.10. Фирма должна разработать агрегированный план производства своей продукции на три месяца. В каждом месяце продукция может выпускаться в нормальное и сверхурочное время. Исходные данные заданы в следующей таблице:

Месяц	Производственная мощность (в единицах)		Стоимость производства единицы продукции		Ожидаемый спрос (в единицах)
	Нормальное время	Сверхурочное время	Нормальное время	Сверхурочное время	
1	100	20	14	18	60
2	100	12	18	21	80
3	65	18	18	21	140

Глава 3
Квадратичное программирование

Будем рассматривать *задачу квадратичного программирования* (КП) следующего вида

$$\begin{aligned} Q(x) = c^T x + \frac{1}{2} x^T D x \to \min, \\ Ax \geq b, \\ x \geq 0, \end{aligned} \tag{3.1}$$

где A есть матрица размера $m \times n$, D — симметричная матрица размера $n \times n$, $c \in \mathbb{R}^n$, $b \in \mathbb{R}^m$. Если D не симметрична, то нужно записать $Q(x)$ в виде $Q(x) = c^T x + \frac{1}{2} x^T \bar{D} x$, где $\bar{D} = \frac{1}{2}(D + D^T)$.

3.1. Критерий оптимальности

Запишем необходимые условия оптимальности Куна-Таккера для задачи (3.1):

$$\begin{aligned} c + Dx - A^T y - u = 0, \\ y \geq 0,\ u \geq 0, \\ y^T (Ax - b) = 0,\ u^T x = 0, \\ Ax - b \geq 0,\ x \geq 0. \end{aligned} \tag{3.2}$$

Заметим, что если $Q(x)$ — выпуклая функция (D — неотрицательно определенная матрица), то условие (3.2) является также и достаточным.

Вводя переменные избытка $v = Ax - b$, перепишем (3.2) в следующем виде:

$$\begin{bmatrix} u \\ v \end{bmatrix} - \begin{bmatrix} D & -A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} c \\ -b \end{bmatrix},$$
$$\begin{bmatrix} u \\ v \end{bmatrix} \geq 0, \quad \begin{bmatrix} x \\ y \end{bmatrix} \geq 0, \quad \text{и} \quad \begin{bmatrix} u \\ v \end{bmatrix}^T \begin{bmatrix} x \\ y \end{bmatrix} = 0. \tag{3.3}$$

Задача (3.3) является частным случаем линейной задачи о дополнительности.

3.2. Линейная задача о дополнительности

Линейная задача о дополнительности обобщает задачи линейного и квадратичного программирования, биматричные игры и еще много других задач.

Пусть M есть квадратная матрица размера n, а вектор $q \in \mathbb{R}^n$. В *линейная задаче о дополнительности* (ЛЗД) нужно найти векторы $w = (w_1, \ldots, w_n)$ и $z = (z_1, \ldots, z_n)$, удовлетворяющие следующим условиям:

$$w - Mz = q, \tag{3.4a}$$
$$w^T z = 0, \tag{3.4b}$$
$$w, z \geq 0. \tag{3.4c}$$

3.2.1. Алгоритм Лемке

Допустимый базис для (3.4), в котором базисной является точно одна переменная из каждой дополняющей пары (w_j, z_j), называется *дополняюще-допустимым.* Алгоритм начинает работу с почти дополняюще-допустимого базиса (это понятие варьируется в зависимости от решаемой задачи) и заканчивает работу, как только будет получен дополняюще-допустимый базис. На каждой итерации, за исключением 1-й, на которой строится начальный почти дополняюще-допустимый базис, вычисления проводятся по следующим правилам:

а) (*правило о дополнительности*) в базис всегда вводится дополнение переменной, покинувшей базис на предыдущей итерации;

б) выбор переменной, покидающей базис, и пересчет таблицы осуществляются по тем же правилам, что и в симплекс-методе.

3.2.2. Пример

Решим следующую задачу квадратичного программирования:

$$\begin{aligned} 6x_1 + 3x_2 - \frac{1}{2}x_1^2 - x_1x_2 - x_2^2 &\to \max, \\ x_1 + x_2 &\le 4, \\ x_1 &\le 2, \\ x_1, x_2 &\ge 0. \end{aligned} \tag{3.5}$$

Сначала перепишем задачу в виде (3.1):

$$\begin{aligned} -6x_1 - 3x_2 + \frac{1}{2}(x_1^2 + 2x_1x_2 + 2x_2^2) &\to \min, \\ -x_1 - x_2 &\ge -4, \\ -x_1 &\ge -2, \\ x_1, x_2 &\ge 0. \end{aligned}$$

Здесь

$$c = \begin{bmatrix} -6 \\ -3 \end{bmatrix}, \quad b = \begin{bmatrix} -4 \\ -2 \end{bmatrix},$$
$$D = \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix}, \quad A = \begin{bmatrix} -1 & -1 \\ -1 & 0 \end{bmatrix}.$$

Сразу заметим, поскольку матрица D положительно определена, то локальный оптимум в задаче (3.5) является также и глобальным оптимумом. Теперь запишем условия оптимальности как ЛЗД вида (3.3):

$$\begin{gathered} \begin{bmatrix} u_1 \\ u_2 \\ v_1 \\ v_2 \end{bmatrix} - \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 2 & 1 & 0 \\ -1 & -1 & 0 & 0 \\ -1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} -6 \\ -3 \\ 4 \\ 2 \end{bmatrix}, \\ u_1, u_2, x_1, x_2, y_1, y_2, v_1, v_2 \ge 0, \\ u_1x_1 = 0, \ u_2x_2 = 0, \ v_1y_1 = 0, \ v_2y_2 = 0. \end{gathered} \tag{3.6}$$

Перепишем задачу (3.6) в табличной форме:

Базис	q	u_1	u_2	v_1	v_2	x_1	x_2	y_1	y_2
u_1	-6	1	0	0	0	-1	-1	-1	-1
u_2	-3	0	1	0	0	-1	-2	-1	0
v_1	4	0	0	1	0	1	1	0	0
v_2	2	0	0	0	1	1	0	0	0

Единственное отличие данной таблицы от таблицы симплекс-метода в том, что здесь отсуиствует строка $-z$ приведенных стоимостей. Как и в симплекс-методе базисным переменным соответствуют единичные столбцы симплексной таблицы.

Итерация 1. Чтобы построить начальный почти дополняюще-допустимый базис, нужно всести дополнительный столбец, соответствующий новой переменной s:

Базис	q	u_1	u_2	v_1	v_2	x_1	x_2	y_1	y_2	s
u_1	-6	1	0	0	0	-1	-1	-1	-1	-1
u_2	-3	0	1	0	0	-1	-2	-1	0	-1
v_1	4	0	0	1	0	1	1	0	0	-1
v_2	2	0	0	0	1	1	0	0	0	-1

Выбираем строку $t = 1$ с минимальным элементом $q_t = -6$ и выполняем операцию замещения с ведущими столбцом s и строкой 1, соответствующей базисной переменной u_1, которая должна покинуть базис. В результате получим следующую таблицу

Базис	q	u_1	u_2	v_1	v_2	x_1	x_2	y_1	y_2	s	Отношения
s	6	-1	0	0	0	1	1	1	1	1	$\frac{6}{1} = 6$
u_2	3	-1	1	0	0	0	-1	0	1	0	∞
v_1	10	-1	0	1	0	2	2	1	1	0	$\frac{10}{2} = 5$
v_2	8	-1	0	0	1	2	1	1	1	0	$\frac{8}{2} = 4$ (min)

В данный момент не обращайте внимания на появившийся в новой таблице столбец "Отношения"и выделение столбца x_1 и строки 4. Все это относится к следующей итерации.

Мы называем базис *почти дополняюще-допустимым*, если

а) из каждой дополнительной пары в базис входит не более одной переменной;

б) точно одна дополняющая пара не представлена в базисе (обе переменные не входят в базис);

в) переменная s входит в базис.

В нашем случае в базисе не представлена пара (u_1, x_1).

Итерация 2. Согласно правилу о дополнительности в базис нужно вводить переменную x_1 (дополнение переменной u_1, покинувшей базис на предыдущей итерации). Далее действуем точно также, как и в симплекс-методе. Чтобы определить переменную для вывода из базиса, вычисляем отношения элементов столбцов q и x_1. Среди этих отношений выбираем минимальный элемент 4, который лежит в строке 4 (v_2 покинет базис). Выполняя замещение с ведущими строкой 4 и столбцом x_1, получим следующую таблицу

Базис	q	u_1	u_2	v_1	v_2	x_1	x_2	y_1	y_2	s	Отношения
s	2	$-\frac{1}{2}$	0	0	$-\frac{1}{2}$	0	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	1	$\frac{2}{1/2} = 4$
u_2	3	-1	1	0	0	0	-1	0	1	0	$\frac{3}{1} = 3$ (min)
v_1	2	0	0	1	-1	0	1	0	0	0	∞
x_1	4	$-\frac{1}{2}$	0	0	$\frac{1}{2}$	1	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	0	$\frac{4}{1/2} = 8$

Итерация 3. На предыдущей итерации из базиса вышла переменная v_2, поэтому ее дополнение y_2 нужно вводить в базис. Вычисляем отношения элементов столбцов q и y_2. Минимальное из этих отношений лежит в строке 2, соответствующей базисной переменной u_2, которую нужно выводить из базиса. Выполняя замещение с ведущими строкой 2 и столбцом y_2, получим следующую таблицу

Базис	q	u_1	u_2	v_1	v_2	x_1	x_2	y_1	y_2	s	Отношения
s	$\frac{1}{2}$	0	$-\frac{1}{2}$	0	$-\frac{1}{2}$	0	1	$\frac{1}{2}$	0	1	$\frac{1}{2}$ (min)
y_2	3	-1	1	0	0	0	-1	0	1	0	∞
v_1	2	0	0	1	-1	0	1	0	0	0	$\frac{2}{1} = 2$
x_1	$\frac{5}{2}$	0	$-\frac{1}{2}$	0	$\frac{1}{2}$	1	1	$\frac{1}{2}$	0	0	$\frac{5}{2}$

Итерация 4. На предыдущей итерации из базиса вышла переменная u_2, поэтому ее дополнение x_2 нужно вводить в базис. Вычисляем отношения элементов столбцов q и x_2. Минимальное из этих отношений лежит в строке 1, соответствующей базисной переменной s, которую нужно выводить из базиса. Выполняя замещение с ведущими строкой 1 и столбцом s, получим следующую таблицу

Базис	q	u_1	u_2	v_1	v_2	x_1	x_2	y_1	y_2	s
x_2	$\frac{1}{2}$	0	$-\frac{1}{2}$	0	$-\frac{1}{2}$	0	1	$\frac{1}{2}$	0	1
y_2	$\frac{7}{2}$	-1	$\frac{1}{2}$	0	$-\frac{1}{2}$	0	0	$\frac{1}{2}$	1	1
v_1	$\frac{3}{2}$	0	$\frac{1}{2}$	1	$-\frac{1}{2}$	0	0	$-\frac{1}{2}$	0	-1
x_1	2	0	0	0	1	1	0	0	0	-1

Как только базис покинула переменная s, мы получаем дополняюще-допустимый базис с ненулевыми базисными компонентами $v_1 = 3/2$, $x_1 = 2$, $x_2 = 1/2$ и $y_2 = 7/2$.

Ответ: $x = (2, 1/2)$ — оптимальное решение задачи (3.5).

3.3. Модель Марковица оптимизации портфеля

Х. Марковиц (H. Markowitz) получил Нобелевскую премию 1990 года в области экономики за его *модель оптимизации портфеля* [5], в которой возврат портфеля — это случайная величина, а риск определяется как дисперсия (вариация) этой случайной величины.

Мы хотим инвестировать сумму B в некоторые из n активов (акций). Пусть p_i есть относительное изменение цены актива i в течении планового периода, т. е. p_i есть изменение цены актива за плановый период, деленное на его цену в начале периода (возврат на один вложенный рубль). Портфелем называют вектор $x = (x_1, \ldots, x_n)^T$, где x_i есть сумма (в рублях или долларах), инвестированная в актив i $(i = 1, \ldots, n)$, $\sum_{i=1}^n x_i = B$, Если $x_i \geq 0$, то мы имеем нормальную длинную позицию в актив i; если $x_i < 0$, то мы имеем короткую позицию (т. е. обязательство в течении планового периода купить активы i на сумму $-x_i$) в актив i.

Будем предполагать, что портфель формируется, чтобы оставаться неизменным в течение заданного планового периода (например, одного года). Считаем, что $p_1, \ldots, p_n$ есть зависимые нормальные случайные величины, а $p = (p_1, \ldots, p_n)^T$ есть случайный вектор цен с известным средним (матожиданием) $\bar{p}$ и коварицонной матрицей Σ. Поэтому возврат портфеля x в течение планового периода есть случайная величина $p^T x$ со средним значением (матожиданием) $\bar{p}^T x$ и вариацией (дисперсией) $x^T \Sigma x$.

[5] H. Markowitz. *Portfolio Selection: Efficient Diversification of Investments*, **1959**, Wiley, New York.

Х. Марковиц сформулировал задачу оптимизации портфеля как следующую задачу квадратичного программирования:

$$x^T \Sigma x \to \min, \tag{3.7a}$$
$$\bar{p}^T x \geq r_{\min}, \tag{3.7b}$$
$$\sum_{i=1}^{n} x_i \leq B, \tag{3.7c}$$
$$x_i \geq 0, \quad i = 1, \ldots, n. \tag{3.7d}$$

В этой задаче мы минимизируем риск портфеля, который Х. Марковиц определил равным вариации портфеля, при гарантированном среднем возврате $r_{\min}$ (ограничение (3.7b)). Естественно, что мы требуем выполнения бюджетного ограничения (3.7c). В базовой модели короткие позиции не допускаются (ограничения (3.7d)).

Чтобы разрешить короткие позиции, в модели (3.7) неравенства (3.7d) нужно заменить следующей системой:

$$x_i^l \geq 0,\ x_i^s \geq 0,\ x_i = x_i^l - x_i^s, \quad i = 1, \ldots, n, \tag{3.8a}$$
$$\sum_{i=1}^{n} x_i^s \leq \eta \sum_{i=1}^{n} x_i^l\,. \tag{3.8b}$$

Здесь неравенство (3.8b) ограничивает общий объем коротких позиций долей η (например, $\eta = 0.25$) от общего объема длинных позиций.

В качестве еще одного расширения базовой модели, введем в нее линейные стоимости транзакций. Начиная с заданного начального портфеля x^0, мы покупаем и продаем активы, чтобы сформировать портфель x, который не будет меняться до конца планового периода. Мы знаем стоимости покупки f_i^b и продажи f_i^s доли актива i стоимостью в 1 ($i = 1, \ldots, n$). Вводя для каждого актива i две новые переменные y_i^b и y_i^s, которые определяют объемы покупки и продажи этого актива при формировании портфеля x, мы добавляем к модели следующие ограничения:

$$x_i = x_i^0 + y_i^b - y_i^s,\ y_i^b \geq 0,\ y_i^s \geq 0, \quad i = 1, \ldots, n.$$

Кроме этого, бюджетное ограничение $\sum_{i=1}^{n} x_i \leq B$ мы заменяем равенством

$$\sum_{i=1}^{n} (1 + f_i^b) y_i^b = B + \sum_{i=1}^{n} (1 - f_i^s) y_i^s\,,$$

которое означает, что количество денег, потраченное на покупку новых активов, должна быть равно количеству денег B (возможно, что $B = 0$), выделенной на формирование нового портфеля, плюс количество денег, полученных от продажи активов.

3.3.1. Пример

Рассмотрим задачу формирования портфеля из 4-х активов без коротких позиций. Средние относительные изменения цен активов и стандартные отклонения представлены в следующей таблице

Актив	$\bar{p}_i$	σ_i
1	0.12	0.2
2	0.1	0.1
3	0.07	0.05
4	0.03	0

Здесь актив 4 — это безрисковый актив с возвратом 3 %. Коэффициенты корреляции между рискованными активами следующие:

$$\rho_{12} = 0.03, \;\; \rho_{13} = -0.04 \text{ и } \rho_{23} = 0.$$

Используя равенства $\Sigma_{ii} = \sigma_i^2$, $\Sigma_{ij} = \rho_{ij}\sigma_i\sigma_j$ и тот факт, что $\sigma_4 = 0$, вычислим коварицонную матрицу:

$$\Sigma = \begin{bmatrix} 0.04 & 0.0006 & -0.0004 & 0 \\ 0.0006 & 0.01 & 0 & 0 \\ -0.0004 & 0 & 0.0025 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Теперь мы можем записать задачу (3.7):

$$\begin{aligned} &0.04x_1^2 + 0.01x_2^2 + 0.0025x_3^2 + 0.0012x_1x_2 - 0.0008x_1x_3 \to \min, \\ &0.12x_1 + 0.1x_2 + 0.07x_3 + 0.03x_4 \geq r_{\min}, \\ &x_1 + x_2 + x_3 + x_4 = 1, \\ &x_1 \geq 0, \; x_2 \geq 0, \; x_3 \geq 0, \; x_4 \geq 0. \end{aligned}$$

Заметим, что, полагая $B = 1$, мы вычислим долю x_i средств, вложенных в актив i $(i = 1, \ldots, n)$.

3.4. Регрессия с ограничениями на коэффициенты

Чтобы приспособить свои учебные программы к потребностям практики, экономический факультет университета решил определить долю своих студентов, работающих в различных областях народного хозяйства.

В результате опроса студентов прошлых выпусков получены следующие данные:

- N_t — количество выпускников в году $t = 1, \dots, T$;
- q_{it} количество выпускников года t, работающих в области i, $i = 1, \dots, m$, $t = 1, \dots, T$.

Чтобы по полученным данным предсказать долю λ_i будущих выпускников, которые будут работать в отрасли $i = 1, \dots, m$, можно воспользоваться методом наименьших квадратов и решить следующую задачу КП:

$$\begin{aligned}
&\sum_{t=1}^{T}\sum_{i=1}^{m}(q_{it} - \lambda_i N_t)^2 \to \min,\\
&\sum_{i=1}^{m}\lambda_i = 1,\\
&\lambda_i \ge 0, \quad i = 1, \dots, m.
\end{aligned}$$

3.5. Аппроксимация выпуклыми функциями

Сначала рассмотрим *задачу интерполяции выпуклыми функциями.*

Теорема 3.1. *Выпуклая функция, которая в точках $x^1, \dots, x^m \in \mathbb{R}^n$ принимает заданные значения $y_1, \dots, y_m \in \mathbb{R}$, существует тогда и только тогда, когда существуют векторы $g^1, \dots, g^m \in \mathbb{R}^n$, которые удовлетворяют системе неравенств*

$$(x^j - x^i)^T g^i \le y_j - y_i, \quad i, j = 1, \dots, m. \tag{3.9}$$

Если векторы $g^1, \dots, g^m$ образуют решение системы (3.9), то

$$f(x) \stackrel{\text{def}}{=} \max_{1\le i\le m}\left(y_i + (g^i)^T(x - x^i)\right)$$

есть искомая выпуклая функция.

Доказательство. Пусть f искомая выпуклая функция, определенная на $\mathbb{R}^n$. По теореме A.5, в каждой точке x^i существует субградиент $g^i \in \mathbb{R}^n$ функции f, что

$$f(x) - f(x^i) \ge (g^i)^T(x - x^i) \quad \forall\, x \in \mathbb{R}^n,\ i = 1, \dots, m. \tag{3.10}$$

Подставляя вместо x точки x^j, после перегруппировки получим систему (3.9). $\square$

Задача аппроксимации выпуклыми функциями формулируется следующим образом: найти выпуклую функцию f, которая в заданных точках $x^1, \dots, x^m \in \mathbb{R}^n$ принимает значения $y_1 = f(x^1), \dots, y_m = f(x^m)$, которые «близки» к заданным значениям $\bar{y}_1, \dots, \bar{y}_m$. Если в качестве меры близости взять сумму квадратов отклонений значений функции от заданных значений, в силу теоремы 3.1 данная задача аппроксимации сводится к решению следующей задачи квадратичного программирования:

$$\begin{aligned} &\sum_{i=1}^{m} (y_i - \bar{y}_i) \to \min, \\ &(x^j - x^i)^T g^i \le y_j - y_i, \quad i, j = 1, \dots, m. \end{aligned} \tag{3.11}$$

Заметим, что в задаче (3.11) неизвестными являются $y_1, \dots, y_m$ и компоненты векторов $g^1, \dots, g^m$, всего $m(n+1)$ неизвестных.

3.6. Назначение цен на молочную продукцию

Правительство страны должно определиться с ценами на молочную продукцию: молоко, масло и сыр. Все эти продукты вырабатываются из производимого в стране сырого молока. Удобно считать, что все сырое молоко разделяется на две компоненты: жиры и сухое молоко, которые позже используются для производства молока, масла и двух видов сыров. После вычитания доли жиров и сухого молока, которые используются для производства продуктов на экспорт и потребления на фермах, для производства продуктов для внутреннего потребления остается 600 000 тон жиров и 750 000 тон сухого молока.

Процентный состав четырех молочных продуктов следующий:

Продукт	Жиры	Сухое молоко	Вода
Молоко	4	9	87
Масло	80	2	18
Сыр 1	35	30	35
Сыр 2	25	40	35

В предыдущем году объемы внутреннего потребления и цены на продукты были следующие.

	Молоко	Масло	Сыр 1	Сыр 2
Потребление (1000 тон)	4820	320	210	70
Цена ($ за тонну)	594	1440	2100	1630

Эластичность E_A спроса на некоторый продукт A при изменении его цены определяется как отношение процентного уменьшения спроса к процентному увеличению цены. Для взаимозаменяемых продуктов A и B крос-эластичность E_{AB} спроса на продукт A при изменении цены на продукт B определяется как отношение процентного увеличение спроса на продукт A к процентному увеличению цены на продукт B. Для наших четырех продуктов эластичности и крос-эластичности были вычислены по историческим данным, и результат представлен в следующей таблице.

Молоко E_M	Масло E_B	Сыр 1 E_{C_1}	Сыр 2 E_{C_2}	Сыр 1/Сыр 2 E_{C_1,C_2}	Сыр 2/Сыр 1 E_{C_2,C_1}
0.4	2.7	1.1	0.4	0.1	0.4

Правительство хочет назначить такие цены, чтобы максимизировать прибыль от реализации молочных продуктов на внутреннем рынке, при дополнительном политическом ограничении, что индех цен на молочную продукцию не должен вырасти, т. е. новые цены должны быть такими, что при новых ценах общая стоимость потребленного в предшествующий год, не дожна превосходить общей стоимости потребленного в тот год, но вычисленной по существовавшим тогда ценам. Какова (экономическая) цена этого политического ограничения?

3.6.1. Формулировка

Пусть $x_M, x_B, x_{C_1}, x_{C_2}$ и $p_M, p_B, p_{C_1}, p_{C_2}$ есть соответственно потребление (= производству) (в тысячах тон) и цены (в тысячах долларов за тонну) на молоко, масло, сыр 1 и сыр 2.

$$p_M x_M + p_B x_B + p_{C_1} x_{C_1} + p_{C_2} x_{C_2} \to \max, \tag{3.12a}$$
$$0.04 x_M + 0.8 x_B + 0.35 x_{C_1} + 0.25 x_{C_2} \le 600, \tag{3.12b}$$
$$0.09 x_M + 0.02 x_B + 0.3 x_{C_1} + 0.4 x_{C_2} \le 750, \tag{3.12c}$$
$$4.82 p_M + 0.32 p_B + 0.21 p_{C_1} + 0.07 p_{C_2} \le 3.87898, \tag{3.12d}$$
$$\frac{\mathrm{d}x_M}{x_M} = -E_M \frac{\mathrm{d}p_M}{p_M}, \tag{3.12e}$$
$$\frac{\mathrm{d}x_B}{x_B} = -E_B \frac{\mathrm{d}p_B}{p_B}, \tag{3.12f}$$
$$\frac{\mathrm{d}x_{C_1}}{x_{C_1}} = -E_{C_1} \frac{\mathrm{d}p_{C_1}}{p_{C_1}} + E_{C_1,C_2} \frac{\mathrm{d}p_{C_2}}{p_{C_2}}, \tag{3.12g}$$
$$\frac{\mathrm{d}x_{C_2}}{x_{C_2}} = -E_{C_2} \frac{\mathrm{d}p_{C_2}}{p_{C_2}} + E_{C_2,C_1} \frac{\mathrm{d}p_{C_1}}{p_{C_1}}, \tag{3.12h}$$
$$x_M \ge 0,\ x_B \ge 0,\ x_{C_1} \ge 0,\ x_{C_2} \ge 0, \tag{3.12i}$$
$$p_M \ge 0,\ p_B \ge 0,\ p_{C_1} \ge 0,\ p_{C_2} \ge 0. \tag{3.12j}$$

Здесь целевая функция (3.12a) вычисляет стоимость произведенной продукции, которая должна быть максимальной. Неравенства (3.12b) и (3.12c) — это ограничения на ресурсы: жиры и сухое молоко. Неравенство (3.12d) выражает ограничение на индекс цен: стоимость произведенного в новых ценах не дожна превышать стоимости проданного в предшествующем году, которая равна

$$\begin{aligned}&0.594 \cdot 4\,820\,000 + 1.440 \cdot 320\,000 + \\ &2.100 \cdot 210\,000 + 1.630 \cdot 70\,000 = 3\,878\,980 \text{ тыс. долларов.}\end{aligned}$$

Дифференциальные уравнения (3.12e)–(3.12h) выражают отношения эластичности для каждого из четырех продуктов.

Мы могли бы решить дифференциальные уравнения (3.12e)–(3.12h), чтобы выразить x переменные через p переменные. Подставив полученные выражения в целевую функцию (3.12a) и в ограничения (3.12b), (3.12c), мы бы получили задачу нелинейного программирования с нелинейностями в целевой функции и двух ограничениях.

Чтобы избежать появления нелинейностей в ограничениях, можно приближенно выразить дифференциальные равенства следующими разностными равенствами:

$$
\begin{aligned}
&\frac{x_M - \bar{x}_M}{\bar{x}_M} = -E_M \frac{p_M - \bar{p}_M}{\bar{p}_M},\\
&\frac{x_B - \bar{x}_B}{\bar{x}_B} = -E_B \frac{p_B - \bar{p}_B}{\bar{p}_B},\\
&\frac{x_{C_1} \bar{x}_{C_1}}{\bar{x}_{C_1}} = -E_{C_1} \frac{p_{C_1} - \bar{p}_{C_1}}{\bar{p}_{C_1}} + E_{C_1,C_2} \frac{p_{C_2} - \bar{p}_{C_2}}{\bar{p}_{C_2}},\\
&\frac{x_{C_2} - \bar{x}_{C_2}}{\bar{x}_{C_2}} = -E_{C_2} \frac{\mathrm{d}p_{C_2}}{p_{C_2}} + E_{C_2,C_1} \frac{p_{C_1} - \bar{p}_{C_1}}{\bar{p}_{C_1}},
\end{aligned}
\tag{3.13}
$$

где $\bar{x}_M, \bar{x}_B, \bar{x}_{C_1}, \bar{x}_{C_2}$ и $\bar{p}_M, \bar{p}_B, \bar{p}_{C_1}, \bar{p}_{C_2}$ есть соответственно спрос (в тысячах тон) и цены (в тысячах долларов) на молоко, масло, сыр 1 и сыр 2 в предшествующий год. Данная аппроксимация будет достаточно точной, если вычисленные значения x и p не будут существенно отличаться от $\bar{x}$ и $\bar{p}$.

Используя равенства (3.13), выразим x переменные через p переменные и результат подставим в целевую функцию (3.12a) и в ограничения (3.12b), (3.12c):

$$
\begin{aligned}
& -6492p_M^2 - 1200p_B^2 - 220p_{C_1}^2 - 34p_{C_2}^2 + 53p_{C_1}p_{C_2} + && \text{(3.14a)}\\
& 6748p_M + 1184p_B + 420p_{C_1} + 70p_{C_2} \to \max, && \text{(3.14b)}\\
& 260p_M + 960p_B + 70.25p_{C_1} - 0.6p_{C_2} \ge 782, && \text{(3.14c)}\\
& 584p_M + 24p_B + 55.2p_{C_1} + 5.8p_{C_2} \ge 35, && \text{(3.14d)}\\
& 4.82p_P + 0.32p_B + 0.21p_{C_1} + 0.07p_{C_2} \le 3.87898, && \text{(3.14e)}\\
& p_M \le 1.039,\ p_B \le 0.987, && \text{(3.14f)}\\
& 220p_{C_1} - 26p_{C_2} \le 420,\ -27p_{C_1} + 34p_{C_2} \le 70, && \text{(3.14g)}\\
& 0.9\bar{p}_M \le p_M \le 1.1\bar{p}_M,\ 0.9\bar{p}_M \le p_M \le 1.1\bar{p}_M, && \text{(3.14h)}\\
& 0.9\bar{p}_{C_1} \le p_{C_1} \le 1.1\bar{p}_{C_1},\ 0.9\bar{p}_{C_2} \le p_{C_2} \le 1.1\bar{p}_{C2}. && \text{(3.14i)}
\end{aligned}
$$

Здесь неравенства в (3.14f) и (3.14g) выражают неравенства $x_M \ge 0$, $x_B \ge 0$, $x_{C_1} \ge 0$ и $x_{C_2} \ge 0$ в переменных p_M, p_B, p_{C_1} и p_{C_2}. Неравенства (3.14h) и (3.14i) введены для того, чтобы не позволить ценам существенно измениться.

После умножения целевой функции на -1 задача (3.14) превратится в задачу выпуклого квадратичного программирования. Заметим однако, что, если бы эластичности были другими, мы могли бы получить и невыпуклую целевую функцию. В таком случае нам пришлось бы ограничиться поиском локального оптимума, или свести полученную задачу к задаче смешанно-целочисленного программирования, чтобы найти глобальный оптимум.

Чтобы оценить (экономическую) цену политического ограничения на индекс цен, нужно также решить задачу (3.14) без ограничения (3.14e), что приведет к увеличению оптимального значения полученной прибыли. Вот этот прирост прибыли и есть цена данного политического ограничения.

3.7. Упражнения

3.1. *Регуляризация Тихонова.* Рассмотрим ситуацию, когда мы хотим найти небольшой вектор x, для которого и вектор невязок $Ax - b$ также небольшой. *Регуляризация* — это метод скаляризации для решения двухкритериальной задачи $(\|ax - b\|, \|x\|)^T \to \min>$ Если используется евклидова норма $\|\cdot\|_2$, то наиболее используемый метод регуляризации Тихонова сводит данную двукритериальную задачу к минимизации взвешенной суммы квадратов:

$$\|Ax - b\|^2 + \delta\|x\|^2 \to \min. \tag{3.15}$$

Докажите, что задача (3.15) имеет следующее аналитическое решение:

$$x = \left(A^T A + \delta I\right)^{-1} A^T b.$$

3.2. Решить задачу квадратичного программирования

$$\begin{aligned} 2x_1 + x_1^2 + x_2^2 - x_1 x_2 &\to \min, \\ x_1 + x_2 &= 2, \\ x_1 \ge 1,\ x_2 &\ge 0. \end{aligned}$$

3.3. Найти расстояние от точки $x^0 = (1,1)^T$ до многогранника, заданного следующей системой неравенств:

$$\begin{aligned} 2x_1 + x_2 &\le 2, \\ x_2 &\le 1, \\ x_1, x_2 &\ge 0. \end{aligned}$$

Глава 4
Смешанно-целочисленное программирование

Задача *смешанно-целочисленного программирования* (СЦП) есть следующая оптимизационная задача:

$$\max\{c^T x \,:\, b^1 \le Ax \le b^2,\ d^1 \le x \le d^2,\ x_j \in \mathbb{Z} \text{ для } j \in S\}, \tag{4.1}$$

где $b^1, b^2 \in \mathbb{R}^m$, $c, d^1, d^2 \in \mathbb{R}^n$, A — действительная $m \times n$-матрица, x — n-вектор переменных (неизвестных), а $S \subseteq \{1, \dots, n\}$ есть множество целочисленных переменных. В задаче *целочисленного программирования* (ЦП) все переменные целочисленны ($|S| = n$).

Задача СЦП отличается от задачи *линейного программирования* (ЛП) тем, что некоторые переменные могут принимать значения из дискретного множества. Это отличие делает задачу СЦП существенно сложнее с алгоритмической точки зрения. Можно сказать, что задача СЦП — это одна из самых трудных задач математического программирования. И это неудивительно, поскольку многие комбинаторные задачи, включая те, которые считаются самыми трудными, очень просто формулируются как задачи СЦП. Одно из самых распространенных применений СЦП в повседневной жизни касается эффективного использования ограниченных ресурсов.

4.1. Целочисленность и нелинейность

Через условие «x — целое» можно выразить многие нелинейные ограничения. Но сначала мы покажем, что само это ограничение можно записать в непрерывных переменных при гладких ограничениях.

Условие, что x есть *бинарная переменная* (принимает только два значения: 0 и 1), записывается одним квадратичным равенством

$$x^2 - x = 0.$$

Такое представление бинарных переменных позволяет записывать многие задачи комбинаторной оптимизации как задачи квадратичного программирования. Для примера, трудная комбинаторная *задача о разбиении множества*

$$\max\{c^T x : Ax = \mathrm{e},\ x \in \{0,1\}^n\},$$

где $c \in \mathbb{R}^n$, а A есть $m \times n$-матрица с элементами 0 и 1, переписывается как задача квадратичного программирования следующим образом:

$$\begin{aligned} c^T x &\to \max, \\ Ax &= \mathrm{e}, \\ x_i^2 &= x_i\,, \quad i = 1, \ldots, n. \end{aligned}$$

Здесь и далее e обозначает вектор подходящего размера, все компоненты которого равны 1.

Предположим теперь, что целочисленная переменная x неотрицательна и ограничена сверху, т. е. $0 \le x \le d$, где d — положительное целое. Для записи числа d в двоичной системе счисления требуется $k = \lfloor \log d \rfloor + 1$ позиций. Поэтому мы можем представить условие $x \in \{0, 1, \ldots, d\}$ следующей системой уравнений:

$$\begin{aligned} x &= \sum_{i=0}^{k-1} 2^i s_i\,, \\ s_i^2 &= s_i, \quad i = 0, \ldots, k-1. \end{aligned}$$

Итак, мы можем заключить, что задача СЦП сводится к задаче квадратичного программирования и, следовательно, не труднее последней. Но отличительная особенность целочисленного программирования состоит в том, что здесь целочисленность переменных учитывается совершенно особым образом на алгоритмическом уровне посредством ветвления по целочисленным переменным и генерации отсечений.

С практической точки зрения более важным является то, что многие нелинейности моделируются введением целочисленных переменных.

4.1.1. Фиксированные доплаты

Функция стоимости с фиксированными доплатами имеет вид (рис. 4.1)

$$c(x) \stackrel{\text{def}}{=} \begin{cases} ax + b, & \text{если } 0 < l \le x \le u, \\ 0, & \text{если } x = 0. \end{cases}$$

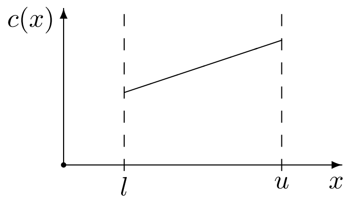


Рис. 4.1. Функция с фиксированными доплатами

Стоимости с фиксированными доплатами появляются всякий раз, когда нельзя пренебречь постоянными издержками, например, стоимостью нового оборудования, затратами на проектирование и т. д.

Если ввести бинарную переменную y и добавить *переменные нижнюю и верхнюю границы* $ly \le x \le uy$, то функцию $c(x)$ можно преобразовать в линейную $\bar{c}(x, y) = ax + by$.

4.1.2. Дискретные переменные

Дискретная переменная x может принимать только конечное число значений $v_1, \dots, v_k$. Например, в задаче проектирования автомобиля объем двигателя x может принимать, скажем, одно из четырех значений: 1.4, 1.6, 1.9 и 2.0 литра.

Дискретную переменную x можно представить как обычную непрерывную переменную, вводя бинарные переменные $y_1, \dots, y_k$ и записывая ограничения

$$x - v_1 y_1 - v_2 y_2 - \dots - v_k y_k = 0, \tag{4.2a}$$

$$y_1 + y_2 + \dots + y_k = 1, \tag{4.2b}$$

$$y_i \in \mathbb{Z}_+, \quad i = 1, \dots, k. \tag{4.2c}$$

В СЦП ограничение типа (4.2b), все переменные которого бинарные, называют *обобщенной верхней границей*. Заметим, что характерной чертой обобщенной верхней границы является то, что только одна ее переменная может принимать ненулевое значение. Обобщенные верхние границы в программной документации часто называют *специальными упорядоченными множествами типа* 1 (SOS1 — Special Ordered Set of Type 1).

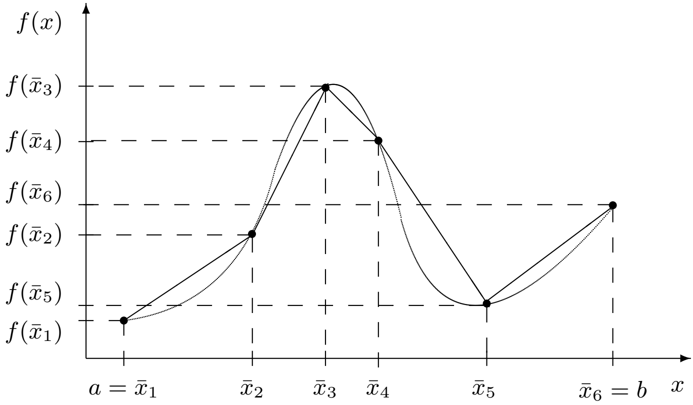


Рис. 4.2. Кусочно-линейная аппроксимация нелинейной функции

4.1.3. Аппроксимация нелинейной функции

Пусть нелинейная функция $y = f(x)$ задана на отрезке $[a, b]$. Выберем некоторое разбиение

$$a = \bar{x}_1 < \bar{x}_2 < \cdots < \bar{x}_r = b$$

отрезка $[a, b]$. Соединяя точки $(\bar{x}_k, \bar{y}_k = f(\bar{x}_k))$ и $(\bar{x}_{k+1}, \bar{y}_{k+1} = f(\bar{x}_{k+1}))$ отрезками прямых, мы получим кусочно-линейную аппроксимацию $\tilde{f}(x)$ функции $f(x)$ (рис. 4.2), которая представляется следующей системой ограничений:

$$x = \sum_{k=1}^{r} \lambda_k \bar{x}_k, \tag{4.3a}$$

$$y = \sum_{k=1}^{r} \lambda_k \bar{y}_k, \tag{4.3b}$$

$$1 = \sum_{k=1}^{r} \lambda_k, \tag{4.3c}$$

$$\lambda_k \le \delta_k, \quad k = 1, \ldots, r, \tag{4.3d}$$

$$1 \ge \delta_i + \delta_j, \quad j = 3, \ldots, r;\ i = 1, \ldots, j-2, \tag{4.3e}$$

$$\lambda_k \ge 0,\ \delta_k \in \{0, 1\}, \quad k = 1, \ldots, r. \tag{4.3f}$$

Здесь ограничения (4.3d)–(4.3f) требуют, чтобы выполнялось условие (SOS2): не более двух переменных λ_k принимают ненулевые значения, причем индексы этих ненулевых переменных должны быть последовательными числами.

Следует отметить, что большинство современных коммерческих библиотек СЦП учитывают условие (SOS2) алгоритмически, организуя ветвление специальным образом. При этом ограничения (4.3d)–(4.3f) в явном виде задавать не нужно, а достаточно только указать, что равенство (4.3c) имеет тип SOS2 (Special Ordered Set of Type 2).

4.1.4. Аппроксимация выпуклой функции

Если функция $f(x)$ выпуклая, то во многих случаях мы можем представить зависимость $y = f(x)$ без введения целочисленных переменных. Как и ранее, аппроксимируем функцию f на интервале определения $[a, b]$ кусочно-линейной функцией $\tilde{f}$ с точками перегиба $a = \bar{x}_1 < \bar{x}_2 < \cdots < \bar{x}_r = b$ (рис. 4.3).

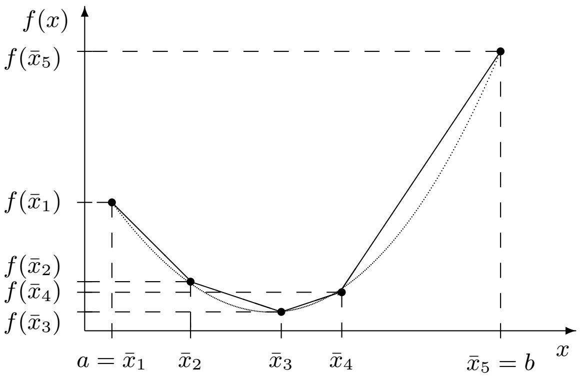


Рис. 4.3. Аппроксимация выпуклой функции

Для $k = 1, \ldots, r-1$ определим числа

$$d_k = \bar{x}_{k+1} - \bar{x}_k,$$
$$q_k = \frac{f(\bar{x}_{k+1}) - f(\bar{x}_k)}{d_k}.$$

В силу выпуклости функции f имеем $q_1 \le q_2 \le \cdots \le q_{r-1}$. Введем дополнительные действительные переменные x_k $(k = 1, \dots, r-1)$ и представим

$$
\begin{aligned}
& x = \sum_{k=1}^{r-1} x_k, \\
& y = f(a) + \sum_{k=1}^{r-1} q_k x_k, \\
& 0 \le x_k \ \le \ d_k, \quad k = 1, \dots, r-1.
\end{aligned}
\tag{4.4}
$$

Нетрудно убедиться в справедливости следующего утверждения.

Утверждение 4.1. *Если коэффициенты при переменной y положительны в ограничениях со знаком «$\le$» и отрицательны в ограничениях со знаком «$\ge$» и в целевой функции (предполагается, что целевая функция должна максимизироваться), то представление* (4.4) *достаточно для выражения зависимости $y = \tilde{f}(x)$.*

4.1.5. Логические условия

Формально мы записываем логические условия с помощью булевых переменных и формул. *Булева переменная* может принимать только два значения: **истина** и **ложь**. Из булевых переменных с помощью бинарных логических операций $\vee$ (*или*), $\wedge$ (*и*) и унарной операции $\neg$ ($\neg x$ означает *не* x) можно образовывать *булевы формулы* почти так же, как из действительных переменных с помощью арифметических операций можно образовывать алгебраические выражения. Например,

$$
(x_1 \vee \neg x_2) \wedge (\neg x_1 \vee x_3) \tag{4.5}
$$

есть булева формула. Подставляя значения для булевых переменных, мы можем вычислить значение булевой формулы, используя правила, представленные в табл. 4.1 и 4.2.

Например, для набора истинности

$$
(x_1, x_2, x_3) = (\textbf{истина}, \textbf{ложь}, \textbf{ложь})
$$

булева формула (4.5) принимает значение **ложь**.

Любую булеву формулу n булевых переменных можно представить в виде *конъюнктивной нормальной формы* (КНФ):

$$
\bigwedge_{i=1}^{m} \left(\bigvee_{j \in S_i} x_j^{\sigma_j^i} \right), \tag{4.6}
$$

Таблица 4.1

Логическая операция $\neg$

x	$\neg x$
ложь	**истина**
истина	**ложь**

Таблица 4.2

Логические операции $\wedge$ и $\vee$

x_1	x_2	$x_1 \wedge x_2$	$x_1 \vee x_2$
ложь	**ложь**	**ложь**	**ложь**
истина	**ложь**	**ложь**	**истина**
ложь	**истина**	**ложь**	**истина**
истина	**истина**	**истина**	**истина**

где $S_i \subseteq \{1, \ldots, n\}$ $(i = 1, \ldots, m)$ и все $\sigma_j^i \in \{0, 1\}$. Здесь мы использовали следующие обозначения: $x^1 \stackrel{\text{def}}{=} x$ и $x^0 \stackrel{\text{def}}{=} \neg x$. Заметим, что булева формула (4.5) представлена в виде КНФ.

КНФ (4.6) принимает значение **истина** тогда и только тогда, когда каждый дизъюнкт $\left(\bigvee_{j \in S_i} x_j^{\sigma_j^i}\right)$ содержит хотя бы один литерал (литералом называется переменная или ее отрицание) со значением **истина**. Если отождествить значение **ложь** с 0, а значение **истина** с 1, то операция отрицания $\neg$ превращает x в $1 - x$. С учетом сказанного наборы истинности, на которых КНФ (4.6) принимает значение **истина**, являются решениями следующей системы неравенств:

$$\begin{gathered} \sum_{j \in S_i^1} x_j + \sum_{j \in S_i^0} (1 - x_j) \geq 1, \quad i = 1, \ldots, m, \\ x_j \in \{0, 1\}, \quad j = 1, \ldots, n. \end{gathered} \tag{4.7}$$

Здесь для $\delta \in \{0, 1\}$ мы использовали обозначение $S_i^\delta \stackrel{\text{def}}{=} \{j \in S_i : \sigma_j^i = \delta\}$. Например, КНФ

$$(x_1 \vee x_2 \vee x_3) \wedge (x_1 \vee \neg x_2) \wedge (x_2 \vee \neg x_3) \wedge (x_3 \vee \neg x_1)$$

принимает значение **истина** на наборах, которые являются решениями системы

$$\begin{aligned} x_1 + x_2 + x_3 &\ge 1, \\ x_1 + (1 - x_2) &\ge 1, \\ x_2 + (1 - x_3) &\ge 1, \\ x_3 + (1 - x_1) &\ge 1, \\ x_1,\ x_2,\ x_3 &\in \{0, 1\}. \end{aligned}$$

4.2. Множественные альтернативы и дизъюнкции

Требуется, чтобы из m неравенств

$$A_i x \le b_i\,, \quad i = 1, \ldots, m,$$

выполнялись не менее q любых неравенств (не важно каких). Например, если два задания i и j должны выполняться на одной машине, то мы должны потребовать выполнения следующей дизъюнкции:

$$e_i - s_j \le 0 \quad \text{или} \quad e_j - s_i \le 0,$$

где s_i и e_i есть соответственно время начала и завершения задания i.

Вводя бинарные переменные

$$y_i = \begin{cases} 1, & \text{если ограничение } A_i x \le b_i \text{ выполняется,} \\ 0 & \text{в противном случае,} \end{cases}$$

мы можем учесть требуемое условие следующим образом:

$$\begin{aligned} A_i x \le b_i + M(1 - y_i)\,, &\quad i = 1, \ldots, m, \\ \sum_{i=1}^{m} y_i \ge q, & \\ y_i \in \{0, 1\}, &\quad i = 1, \ldots, k. \end{aligned}$$

Здесь M — достаточно большое число, такое, что неравенства $A_i x \le b_i + M$ выполняются автоматически для всех допустимых векторов x решаемой задачи.

В заключение рассмотрим случай, когда из двух условий должно выполняться хотя бы одно:

$$x_1 \geq a \quad \text{или} \quad x_2 \geq b.$$

Например, мы хотим иметь рабочую станцию с $x_1 \geq a$ процессорами или однопроцессорную систему с частотой процессора $x_2 \geq b$.

Если обе переменные x_1 и x_2 неотрицательны, то, вводя бинарную переменную y, требуемую дизъюнкцию можно записать в виде

$$x_1 \geq \ ay, \quad x_2 \geq b(1-y).$$

Далее мы продемонстрируем использование множественных альтернатив и дизъюнкций на трех примерах.

4.2.1. Размещение прямоугольных модулей на чипе

На прямоугольном чипе ширины W и высоты H нужно разместить n прямоугольных модулей, модуль i имеет ширину w_i и высоту h_i.

Выберем систему координат с началом O в левом нижнем углу чипа, осью Ox, направленной влево, и осью Oy, направленной вверх. Пусть пара действительных переменных x_i, y_i определяет координату левого нижнего угла модуля i, $i = 1, \ldots, n$.

Очевидно должны выполняться неравенства

$$\begin{aligned} &0 \leq x_i \leq W - w_i, \qquad i = 1, \ldots, n, \\ &0 \leq y_i \leq H - h_i, \qquad i = 1, \ldots, n. \end{aligned} \tag{4.8}$$

Чтобы два модуля i и j не пересекались, нужно потребовать выполнения хотя бы одного из следующих четырех неравенств:

$$\begin{aligned} x_i + w_i \leq x_j \quad &(i \text{ слева от } j), \\ x_j + w_j \leq x_i \quad &(i \text{ справа от } j), \\ y_i + h_i \leq y_j \quad &(i \text{ ниже } j), \\ y_j + h_j \leq y_i \quad &(i \text{ выше } j). \end{aligned}$$

Вводя четыре бинарные переменные z_{ij}^l, z_{ij}^r, z_{ij}^b и z_{ij}^a, мы можем представить данную дизъюнкцию системой неравенств

$$
\begin{aligned}
x_i + w_i &\le x_j + W(1 - z_{ij}^l),\\
x_j + w_j &\le x_i + W(1 - z_{ij}^r),\\
y_i + h_i &\le y_j + H(1 - z_{ij}^b),\\
y_j + h_j &\le y_i + H(1 - z_{ij}^a),\\
z_{ij}^l + z_{ij}^r + z_{ij}^b + z_{ij}^a &\ge 1.
\end{aligned}
\tag{4.9}
$$

Обычно модули разрешается поворачивать на 90°. Введем бинарную переменную δ_i, которая принимает значение 1, если модуль i поворачивается $(i = 1.\dots,n)$. Теперь ширина и высота модуля i соответственно равны

$$
\begin{aligned}
&(1-\delta_i)w_i + \delta_i h_i,\\
&(1-\delta_i)h_i + \delta_i w_i.
\end{aligned}
$$

С учетом этого ограничения (4.8) и (4.9) переписываются следующим образом:

$$
\begin{aligned}
0 \le x_i \le W - ((1-\delta_i)w_i + \delta_i h_i), &\quad i = 1,\dots,n,\\
0 \le y_i \le H - ((1-\delta_i)h_i + \delta_i w_i), &\quad i = 1,\dots,n,\\
x_i + (1-\delta_i)w_i + \delta_i h_i \le x_j + W(1 - z_{ij}^l), &\quad i = 1,\dots,n-1;\ j = i+1,\dots,n,\\
x_j + (1-\delta_j)w_j + \delta_j h_j \le x_i + W(1 - z_{ij}^r), &\quad i = 1,\dots,n-1;\ j = i+1,\dots,n,\\
y_i + (1-\delta_i)h_i + \delta_i w_i \le y_j + H(1 - z_{ij}^b), &\quad i = 1,\dots,n-1;\ j = i+1,\dots,n,\\
y_j + (1-\delta_j)h_j + \delta_j w_j \le y_i + H(1 - z_{ij}^a), &\quad i = 1,\dots,n-1;\ j = i+1,\dots,n,\\
z_{ij}^l + z_{ij}^r + z_{ij}^b + z_{ij}^a \ge 1, &\quad i = 1,\dots,n-1;\ j = i+1,\dots,n,\\
z_{ij}^l, z_{ij}^r, z_{ij}^b, z_{ij}^a \in \{0,1\}, &\quad i = 1,\dots,n-1;\ j = i+1,\dots,n,\\
\delta_i \in \{0,1\}, &\quad i = 1,\dots,n.
\end{aligned}
$$

4.2.2. Линейная задача о дополнительности

Рассмотрим линейную задачу о дополнительности (3.4) из раздела (3.2). Несмотря на свое название, задача (3.4) — это нелинейная задача, поскольку она содержит нелинейное ограничение (3.4a), которое, в силу неотрицательности векторов w и z, эквивалентно системе

$$w_i z_i = 0, \quad i = 1, \dots, n.$$

Каждое из равенств $w_i z_i = 0$ выражает дизъюнкцию: $w_i = 0$ или $z_i = 0$.

В предположении, что мы знаем верхние границы изменения переменных $w_i \le g_i$ и $z_i \le h_i$[6], можно представить нелинейное равенство $w^T z = 0$ следующей системой:

$$w_i \le g_i x_i\,,\ z_i \le h_i(1 - x_i)\,,\ x_i \in \{0, 1\}, \quad i = 1, \dots, n. \tag{4.10}$$

В результате, мы сводим задачу (3.4) к следующей задаче СЦП:

$$c^T w + p^T z \to \max, \tag{4.11a}$$
$$w - Mz = q, \tag{4.11b}$$
$$w_i \le g_i x_i\,,\ z_i \le h_i(1 - x_i), \quad i = 1, \dots, n, \tag{4.11c}$$
$$w, z \ge 0,\ x \in \{0, 1\}^n. \tag{4.11d}$$

Заметим, что векторы c и p в целевой функции могут быть любыми. В этом заключается одно из преимуществ использования модели СЦП (4.11). Выбирая ссответствующим образом векторы c и p, можно среди решений задачи (3.4) целенаправленно искать решение, обладающее желаемыми дополнительными свойствами.

4.2.3. Квадратичное программирование при линейных ограничениях

Задача квадратичного программирования при линейных ограничениях (3.1) изучалась в главе 3. Задачу (3.1) также можно сначала представить как линейную задачу о дополнительности (3.3), а затем записать эквивалентную задачу СЦП. Но здесь мы рассмотрим другой способ сведения, который не требует знания верхних границ для переменных.

Если точка x есть оптимальное решение задачи (3.1), то, по теореме Куна — Таккера, существует такой вектор $y \in \mathbb{R}^m$, что выполняются

[6] Для целочисленной матрицы M мы можем оценить w_i и z_i исходя из правила Крамера для вычисления компонент решения системы линейных уравнений: $w_i, z_i \le \Delta(A)$ для $i = 1, \dots, n$, где $\Delta(A)$ обозначает максимальный по модулю минор расширенной матрицы ограничений $A = [I - M|q]$. Но с практической точки зрения эти оценки будут слишком грубыми.

следующие условия:

$$
\begin{aligned}
& x \geq 0,\ y \geq 0, \\
& Ax \geq b, \\
& c + Dx - A^T y \geq 0, \\
& y^T (Ax - b) = 0, \\
& \left(c + Dx - A^T y\right)^T x = 0.
\end{aligned}
\tag{4.12}
$$

Если (x, y) есть решение системы (4.12), то точка x называется *стационарной точкой* (или точкой Куна — Таккера) задачи (3.1). Если матрица D неотрицательно определенная, то целевая функция выпуклая, и, следовательно, каждая стационарная точка является оптимальным решением задачи (3.1).

Рассмотрим следующую задачу СЦП:

$$
\begin{aligned}
& z \to \max, \\
& 0 \leq Au - bz \leq \mathrm{e} - \alpha, \\
& 0 \leq Du - A^T v + cz \leq \mathrm{e} - \beta, \\
& 0 \leq u \leq \beta,\ 0 \leq v \leq \alpha, \\
& 0 \leq z \leq 1,\ \alpha \in \{0,1\}^m,\ \beta \in \{0,1\}^n.
\end{aligned}
\tag{4.13}
$$

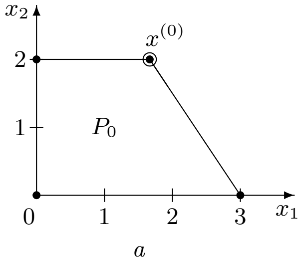
Обозначим через z^* оптимальное значение целевой функции в задаче (4.13). Нетрудно убедиться в справедливости следующих утверждений.

1. Если $z^* = 0$, то задача (3.1) не имеет стационарных точек.
2. Если $z^* > 0$ и (u^*, v^*, z^*) есть оптимальное решение задачи (4.13), то векторы $x^* = (1/z^*)\, u^*$, $y^* = (1/z^*)\, v^*)$ образуют решение системы (4.12), и, значит, x^* есть стационарная точка задачи (3.1).

4.3. Метод сечений

Продемонстрируем суть метода сечений на следующем простом примере:

$$
\begin{aligned}
x_1 + 2x_2 &\to \max, \\
3x_1 + 2x_2 &\leq 9, \\
x_2 &\leq 2, \\
x_1, x_2 &\in \mathbb{Z}_+ .
\end{aligned}
\tag{4.14}
$$



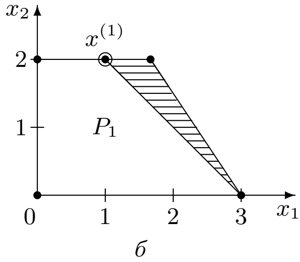


Рис. 4.4. Геометрическая интерпретация метода сечений

Сначала решаем релаксационную задачу ЛП для задачи ЦП (4.14). Напомним, что эта задача получается из исходной задачи ЦП отбрасыванием требования целочисленности переменных. Ее решением является точка $x^{(0)} = (5/3, 2)^T$ (рис. 4.4, *а*). Поскольку эта точка не является целочисленной, то она не является решением задачи (4.14).

Идея метода сечений состоит в том, чтобы «отсечь» точку $x^{(0)}$ от множества

$$X = \{x \in \mathbb{Z}^2_+ :\ 3x_1 + 2x_2 \le 9,\ x_2 \le 2\}$$

допустимых решений задачи (4.14). Это означает, что к ограничениям задачи (4.14) нужно добавить хотя бы одно неравенство, которое нарушается в точке $x^{(0)}$ и которому удовлетворяют все точки из X. Существует несколько способов построить (сгенерировать) нужное неравенство. Некоторые из этих способов мы будем рассматривать позже в этой и следующей главах.

Здесь мы отсечем точку $x^{(0)}$, исходя из простого соображения, что оба неравенства $3x_1 + 2x_2 \le 9$ и $x_2 \le 2$ одновременно не могут выполняться как равенства в точках множества X. Поэтому неравенство

$$3x_1 + 2x_2 + x_2 \le 9 + 2 - 1, \quad \text{или} \quad 3x_1 + 3x_2 \le 10$$

справедливо для X, но не для $x^{(0)}$ $(3 \cdot (5/3) + 3 \cdot 2 = 11 > 10)$. Мы можем усилить полученное неравенство, если сначала разделим его на 3, а затем округлим правую часть:

$$x_1 + x_2 \le \lfloor 10/3 \rfloor = 3.$$

Итак, мы нашли отсечение $x_1 + x_2 \le 3$ и можем добавить его к ограничениям задачи (4.14). В результате от допустимого многогранника P_0 исходной релаксационной задачи ЛП будет отсечена область, которая не содержит точек допустимого множества X (на рис. 4.4, *б* отсеченная область заштрихована). Решением новой релаксационной задачи ЛП с допустимым многогранником P_1 является целочисленная точка $x^{(1)} = (1, 2)^T$, которая и является оптимальным решением задачи ЦП (4.14).

4.4. Метод ветвей и границ

Будем рассматривать задачу СЦП (4.1). Базовой структурой метода ветвей и границ является *дерево поиска. Корень* (или *корневой узел*) дерева поиска соответствует исходной задаче СЦП. В ходе решения задачи дерево растет благодаря процессу, называемому *ветвлением*, который создает двух или более сыновей для одного из листьев текущего дерева поиска. Каждая из задач СЦП в сыновних узлах получается из родительской задачи СЦП добавлением одного или нескольких новых ограничений. Обычно новое ограничение — это верхняя или нижняя граница для переменной. Нужно также заметить, что в процессе ветвления мы не должны потерять допустимые решения: объединение допустимых областей задач сыновей должно давать допустимую область их родителя.

Но если бы дерево поиска только росло (ветвилось), то даже для сравнительно небольших задач его размер мог бы быть огромным. Напротив, одна из главных идей в методе ветвей и границ состоит в том, чтобы не давать дереву поиска разрастаться. Это достигается отсечением «бесперспективных» ветвей дерева поиска. О перспективности узлов дерева поиска мы судим, сравнивая верхние и нижние границы. В методах ветвей и границ, основанных на линейном программировании, *верхней границей* в узле k является оптимальное значение $\gamma(k)$ целевой функции релаксационной задачи ЛП в данном узле. *Нижней границей* (или *рекордом*) называется наибольшее значение R целевой функции на уже найденных допустимых решениях исходной задачи СЦП. Само наилучшее из полученных решений называется *рекордным решением.* Если $\gamma(k) \le R$, то узел k и всех его потомков можно отсечь от дерева поиска.

Базовый вариант метода ветвей и границ для задачи СЦП (4.1) приведен в листинге 4.1. На вход процедуры *branch_and_bound* подаются исходные данные о задаче СЦП (векторы c, b^1, b^2, d^1, d^2, матрица A и множество S), а также допустимое решение x^R и $R = c^T x^R$. Если найти допустимое решение задачи СЦП трудно, то на вход процедуры нужно подавать $R = -\infty$. Если и на выходе процедуры *branch_and_bound* $R = -\infty$, то задача СЦП не имеет допустимых решений. В противном

```
branch_and_bound(c, b¹, b², A, d¹, d², S; xᴿ, R);
{
  Находим x⁰ ∈ arg max{cᵀx : b¹ ≤ Ax ≤ b², d¹ ≤ x ≤ d²};
  if (x⁰_S ∈ ℤ^S) { // изменяем рекорд и рекордное решение
    R = cᵀx⁰; xᴿ = x⁰; return;
  }
  сформировать список активных узлов из одного узла (x⁰, d¹, d²);
  while (список активных узлов непуст) {
    выбрать узел N = (x⁰, d¹, d²) из списка активных узлов;
    if (cᵀx⁰ ≤ R)
      continue;
    выбрать дробную компоненту x⁰_i для i ∈ S;
    найти x¹ ∈ arg max{cᵀx : b¹ ≤ Ax ≤ b², d¹ ≤ x ≤ d²(i, ⌊x⁰_i⌋)};
    if (cᵀx¹ > R) {
      if (x¹_S ∈ ℤ^S) { // изменяем рекорд и рекордное решение
        R = cᵀx¹; xᴿ = x¹;
      }
      else
        добавить узел (x¹, d¹, d²(i, ⌊x⁰_i⌋)) к списку активных узлов;
    }
    найти x² ∈ arg max{cᵀx : b¹ ≤ Ax ≤ b², d¹(i, ⌈x⁰_i⌉)) ≤ x ≤ d²};
    if (cᵀx² > R) {
      if (x²_S ∈ ℤ^S) { // изменяем рекорд и рекордное решение
        R = cᵀx²; xᴿ = x²;
      }
      else
        добавить узел (x², d¹(i, ⌈x⁰_i⌉), d²) к списку активных узлов;
    }
  }
}
```

Листинг 4.1. Метод ветвей и границ для задачи СЦП случае x^R есть оптимальное решение задачи. В описании метода мы используем обозначение

$$d(i,\alpha) \stackrel{\text{def}}{=} \begin{cases} d_j, & \text{если } j \neq i, \\ \alpha, & \text{если } j = i. \end{cases}$$

Из описания процедуры также следует, что в узлах дерева поиска мы храним интервалы изменения переменных и оптимальные решения релаксационных задач ЛП. На практике в каждом узле вместо оптимального решения задачи ЛП нужно хранить описание оптимального базиса, чтобы двойственный симплекс-метод мог быстро решать задачи ЛП для сыновей данного узла.

В методе имеется неоднозначность в выборе узла из списка активных узлов и в выборе дробной переменной, если их несколько. Существует несколько конкурентоспособных стратегий. Простой и в то же время достаточно эффективный способ состоит в том, чтобы выбирать узел с максимальной верхней границей и переменную, дробная часть которой ближе других к 0.5.

Вас не должно смущать то, что в процедуре *branch_and_bound* не упоминается дерево поиска. В действительности, процедура «строит» (хотя и неявно) дерево поиска, причем листья этого дерева и образуют список активных узлов. Продемонстрируем работу метода ветвей и границ на примере.

Пример 4.1. *Решим следующую задачу ЦП*

$$\begin{aligned} & x_1 + 2x_2 \to \max, \\ 1: \quad & -2x_1 + 3x_2 \le 4, \\ 2: \quad & 2x_1 + 2x_2 \le 11, \\ 3: \quad & 1 \le x_1 \le 4, \\ 4: \quad & 1 \le x_2 \le 5, \\ & x_1, x_2 \text{ — целые.} \end{aligned}$$

Решение. Дерево поиска представлено на рис. 4.5. Каждый узел дерева изображен в виде прямоугольника, в котором для релаксационной задачи ЛП в этом узле указаны границы изменения переменных, а также оптимальное значение целевой функции (оценка) и оптимальное решение. Все подзадачи, которые появляются в процессе решения примера методом ветвей и границ, занумерованы числами от 0 (корневой узел, соответствующий исходной задаче) до 5. Релаксационные задачи ЛП для вычисления верхних границ решаются двойственным симплекс-методом

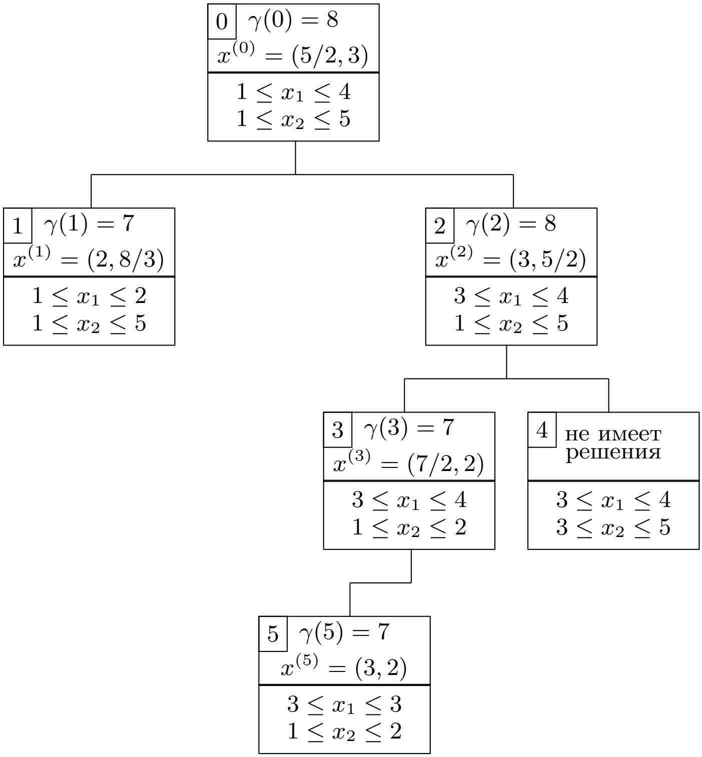


Рис. 4.5. Дерево поиска для задачи ЦП из примера 4.1

начиная с оптимального решения релаксационной задачи ЛП для непосредственного предка. Сначала $R = -\infty$. Поскольку в данном примере на всех допустимых решениях целевая функция принимает только целые значения, то в качестве оценки $\gamma(k)$ мы берем не $c^T x^{(k)}$, а $\lfloor c^T x^{(k)} \rfloor$. Здесь $x^{(k)}$ обозначает оптимальное решение релаксационной задачи в узле k. Ниже представлены все шаги решения задачи. Они занумерованы двумя числами $i.j$, где i есть номер подзадачи, а j — номер итерации двойственного симплекс-метода.

0.0. $I = (3, 4)$, $B^{-1} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$, $x = (4, 5)^T$, $\pi = (1, 2)^T$.

0.1. $s = 1$, $u = (-2, 3)^T$, $\lambda = \frac{2}{3}$, $t = 2$, $I = (3, 1)$,
$B^{-1} = \begin{bmatrix} 1 & 0 \\ 2/3 & 1/3 \end{bmatrix}$, $x = (4, 4)^T$, $\pi = (7/3, 2/3)^T$.

0.2. $s = 2$, $u = (10/3, 2/3)^T$, $\lambda = 7/10$, $t = 1$, $I = (2, 1)$,
$B^{-1} = \begin{bmatrix} 3/10 & -1/5 \\ 1/5 & 1/5 \end{bmatrix}$, $x = x^{(0)} = (5/2, 3)^T$, $\pi = (7/10, 1/5)^T$, $\gamma(0) = 8$.

Поскольку решение $x^{(0)}$ релаксационной задачи ЛП нецелочисленно, то формируем корень (узел 0) дерева поиска и затем осуществляем ветвление по дробной переменной x_1.

1.1. $s = 3$, $u = (3/10, -1/5)^T$, $\lambda = 7/3$, $t = 1$, $I = (3, 1)$,
$B^{-1} = \begin{bmatrix} 1 & 0 \\ 2/3 & 1/3 \end{bmatrix}$, $x^{(1)} = (2, 8/3)^T$, $\pi = (7/3, 2/3)^T$, $\gamma(1) = 7$.

Поскольку и решение $x^{(1)}$ нецелочисленно, а $\gamma(1) = 7 > -\infty = R$, то узел 1 добавляем к дереву поиска.

2.1. $s = -3$, $u = (-3/10, 1/5)^T$, $\lambda = 1$, $t = 2$, $I = (2, -3)$,
$B^{-1} = \begin{bmatrix} 0 & -1 \\ 1/2 & 1 \end{bmatrix}$, $x^{(2)} = (3, 5/2)^T$, $\pi = (1, 1)^T$, $\gamma(2) = 8$.

И в узле 2 решение $x^{(2)}$ нецелочисленно. Так как $\gamma(2) = 8 > -\infty = R$, то этот узел также добавляем к дереву поиска.

Среди активных узлов наибольшая верхняя граница в узле 2. Выполняем ветвление по переменной x_2.

3.1. $s = 4$, $u = (1/2, 1)^T$, $\lambda = 1$, $t = 2$, $I = (2, 4)$,
$B^{-1} = \begin{bmatrix} 1/2 & -1 \\ 0 & 1 \end{bmatrix}$, $x^{(3)} = (7/2, 2)^T$, $\pi = (1/2, 1)^T$, $\gamma(3) = 7$.

В узле 3 решение $x^{(3)}$ нецелочисленно и $\gamma(3) = 7 > -\infty = R$. Этот узел также добавляем к дереву поиска.

4.1. $s = -4$, $u = (-1/2, -1)^T$. Так как все компоненты вектора u неположительны, то задача ЛП не имеет допустимых решений. В этом случае узел 4 не нужно добавлять к дереву поиска. Тем не менее на рис. 4.5 этот узел все же представлен, поскольку с ним рисунок является более информативным.

Из двух активных узлов 1 и 3 с максимальной верхней границей 7 выбираем узел 3 и выполняем ветвление по переменной x_1.

5.1. $s = 3$, $u = (1/2, -1)^T$, $\lambda = 1$, $t = 1$, $I = (3, 4)$,

$$B^{-1} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix},\ x^{(5)} = (3, 2)^T,\ \pi = (1, 2)^T,\ \gamma(5) = 7.$$

Поскольку вектор $x^{(5)}$ целый и $\gamma(5) = 7 > -\infty = R$, то меняем рекорд и рекордное решение: $R = 7$, $x^R = (3, 2)^T$. На рис. 4.5 узел 5 также представлен для большей информативности.

Так как верхние границы для узлов 1 и 3 равны текущему рекорду, то узел 1 и правую ветвь узла 3, которую мы даже не успели сформировать, можно отсечь.

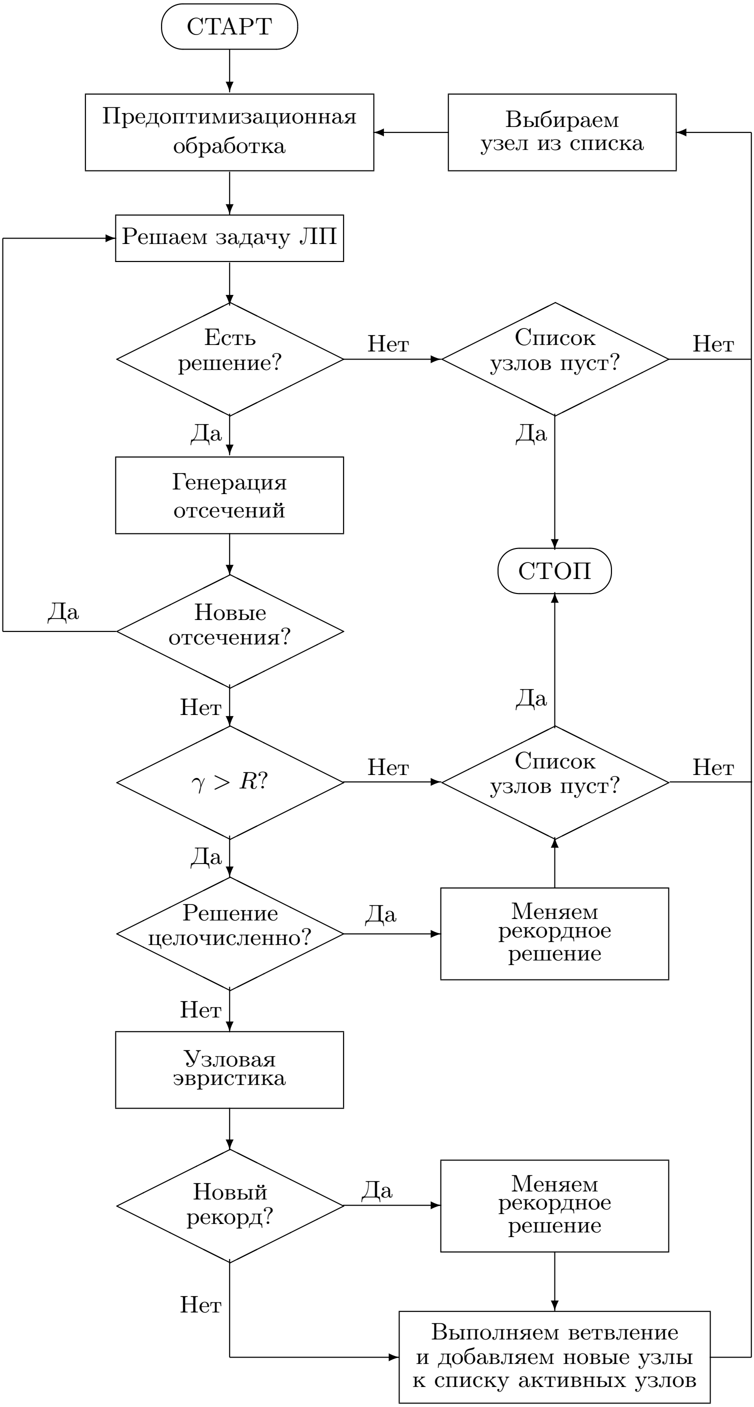
Поскольку в дереве поиска больше нет необработанных узлов (список узлов пуст), то рекордное решение $x^R = (3, 2)^T$ является оптимальным. □

4.5. Метод ветвей и сечений

Метод ветвей и сечений — это метод ветвей и границ, в котором отсечения генерируются при решении релаксационной задачи ЛП во всех (или только некоторых) узлах дерева поиска. На первый взгляд может показаться, что эти изменения незначительны. На практике это изменило всю философию целочисленного программирования. Упрощенная блок-схема метода ветвей и сечений представлена на рис. 4.6.

Две величины определяют поведение метода ветвей и сечений (равно как и метода ветвей и границ) — это нижняя граница (рекорд) и верхние границы (оптимальные значения целевых функций для релаксационных задач ЛП) в узлах дерева поиска. Добавление отсечений способствует уменьшению верхних границ. Если в методе ветвей и границ при обработке очередного узла дерева поиска главной целью было поскорее решить соответствующую ему релаксационную задачу ЛП, то теперь мы выполняем существенно большую работу в каждом узле, генерируя отсечения с целью минимизировать верхнюю границу. При этом *разрыв двойственности* (разность между верхней и нижней границей) в узле также уменьшается.

В отличие от «чистых» методов сечений, мы теперь не надеемся, что одних отсечений будет достаточно для получения оптимального решения. Заметим также, что раньше, как правило, генерировалось только одно неравенство, отсекающее текущее дробное решение. Сегодня такой способ считается плохим, отсечения теперь добавляются группами из многих неравенств.


СТАРТ
Предоптимизационная обработка
Выбираем узел из списка
Решаем задачу ЛП
Есть решение?
Нет
Список узлов пуст?
Нет
Да
Да
Генерация отсечений
СТОП
Да
Новые отсечения?
Да
Нет
$\gamma > R$?
Нет
Список узлов пуст?
Нет
Да
Решение целочисленно?
Да
Меняем рекордное решение
Нет
Узловая эвристика
Новый рекорд?
Да
Меняем рекордное решение
Нет
Выполняем ветвление и добавляем новые узлы к списку активных узлов

На практике очень важно определить момент, когда нужно прекратить генерировать новые отсечения и приступить к ветвлению. Если добавляется много отсечений в каждом узле, то на дооптимизацию узловых задач ЛП может потребоваться существенно большее время. Разумная стратегия состоит в том, чтобы следить за тем, как сокращается разрыв двойственности. Если прогресса нет на протяжении нескольких раундов, то самое время остановиться. При этом после каждого раунда из активной (решаемой в данный момент) задачи ЛП разумно удалять отсечения, которые не выполняются «почти» как равенства. Часть из таких отсечений удаляется из системы насовсем, а другая часть перемещается в специальное хранилище, называемое *пулом отсечений*. Перед тем как добавлять обработанный узел к списку активных узлов, все отсечения активной задачи ЛП, которые еще не были занесены в пул, заносятся туда. Впоследствии, когда данный узел будет выбран из списка активных узлов для выполнения ветвления, извлекая нужные неравенства из пула, мы сможем восстановить задачу ЛП для этого узла.

Может показаться, что пул нужен только для того, чтобы ограничивать размер решаемых задач ЛП. Но если отсечения добавляются не только в корневом узле дерева поиска, то без пула просто не обойтись. Проблема в том, что не все неравенства, генерируемые в методе ветвей и сечений, являются *глобальными*, т. е. справедливыми для всех узлов дерева поиска. Это происходит, в частности, и потому, что при выводе отсечений (например, дробных отсечений Гомори) используются границы для переменных (неравенства вида $x_j \leq (\geq) d$), которые различны в разных узлах дерева поиска. *Локальное неравенство* справедливо для конкретного узла дерева поиска и всех его потомков, а для остальных узлов оно может и не выполняться. Поэтому такое неравенство не может постоянно присутствовать в активной матрице ограничений.

Другой способ сократить разрыв двойственности в узле состоит в применении узловых эвристик с целью увеличить нижнюю границу (рекорд). Идея *узловой эвристики* проста. Прежде чем приступать к ветвлению в конкретном узле, можно попробовать «округлить» оптимальное решение x задачи ЛП в данном узле. Обычно округление состоит в выполнении некоторого типа «ныряния», когда фиксируются значения группы целочисленных переменных с почти целыми значениями, решается полученная задача ЛП, затем фиксируется еще одна группа переменных, и так до тех пор, пока не будет получено целочисленное решение или фиксирование переменных приведет к недопустимости. Если таким образом удастся увеличить нижнюю границу (будет получено целочисленное решение, лучшее рекордного), то это может позволить отсеять

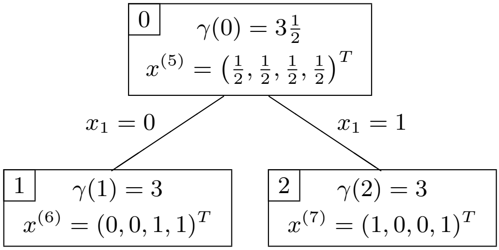


Рис. 4.7. Дерево поиска для примера 4.2

некоторые активные узлы дерева поиска и тем самым ускорить решение всей задачи.

Пример 4.2. *Решим следующую задачу ЦП:*

$$\begin{aligned} x_1 + 3x_2 + \ \ x_3 + 2x_4 &\to \max, \\ 4x_1 + 7x_2 + 3x_3 + 5x_4 &\le 10, \\ 5x_1 + 4x_2 + 6x_3 + 2x_4 &\le 9, \\ x_1, x_2, x_3, x_4 &\in \{0, 1\}. \end{aligned} \tag{4.15}$$

Решение. Давайте условимся генерировать только CI-неравенства. Поскольку эти отсечения являются глобальными, то они верны для всех узлов дерева поиска, представленного на рис. 4.7.

0. Сначала решаем релаксационную задачу ЛП для задачи (4.15). Ее решение есть точка $x^{(1)} = (0, 1, 0, 3/5)^T$. Она не удовлетворяет неравенству

$$x_2 + x_4 \le 1,$$

записанному для покрытия $C_1^1 = \{2, 4\}$ первого рюкзачного неравенства. Добавив это неравенство и выполнив дооптимизацию, получим новое решение $x^{(2)} = = (1/3, 1, 5/9, 0)^T$, которое не удовлетворяет неравенству

$$x_1 + x_2 \le 1$$

для покрытия $C_2^1 = \{1, 2\}$ первого рюкзачного неравенства. Выполняя дооптимизацию, находим решение $x^{(3)} = (0, 1, 5/6, 0)^T$, которое нарушает неравенство

$$x_2 + x_3 \le 1$$

для покрытия $C_1^2 = \{2, 3\}$ второго рюкзачного неравенства. Снова выполняя дооптимизацию, находим решение $x^{(4)} = (5/9, 4/9, 5/9, 5/9)^T$, которое не удовлетворяет неравенству

$$x_1 + x_3 \leq 1$$

для покрытия $C_2^2 = \{1, 3\}$ второго рюкзачного неравенства. Очередная дооптимизация дает следующее решение: $x^{(5)} = (1/2, 1/2, 1/2, 1/2)^T$, которое удовлетворяет всем CI-неравенствам для каждого из двух рюкзачных множеств. Поэтому нужно переходить к ветвлению, которое выполним по переменной x_1.

1. Решаем релаксационную задачу для узла 1 ($x_1 = 0$). Ее целочисленное решение $x^{(6)} = (0, 0, 1, 1)^T$ объявляется в качестве рекордного решения $x^R = (0, 0, 1, 1)^T$, а рекорд устанавливаем равным $R = c^T x^R = 3$.

2. Решение $x^{(7)} = (1, 0, 0, 1)^T$ релаксационной задачи в узле 2 ($x_1 = 1$) также целочисленно. Значение целевой функции на нем также равно 3. Следовательно, обе точки $x^{(6)}$ и $x^{(7)}$ — решения задачи (4.15). □

4.6. Примеры задач СЦП

4.6.1. Потоки с фиксированными доплатами

Транспортная сеть представляется ориентированным графом $G = (V, E)$. Для каждого узла $v \in V$ известна *потребность* d_v в некотором продукте. Если $d_v > 0$, то в узле v имеется *спрос* на данный продукт в объеме d_v; если $d_v < 0$, то в v имеется *предложение* продукта в объеме $-d_v$; для транзитных узлов $d_v = 0$. Предполагается, что спрос и предложение уравновешены: $\sum_{v \in V} d_v = 0$. Пропускная способность дуги $e \in E$ равна $u_e > 0$, а стоимость транспортировки $x_e > 0$ единиц продукта по ней определяется по формуле $f_e + c_e x_e$. Естественно, если продукт по дуге не транспортируется ($x_e = 0$), то платить ничего не нужно. В *задаче о потоке с фиксированными доплатами* нужно определить способ транспортировки продукта из узлов с предложением в узлы со спросом, при котором суммарные транспортные издержки минимальны.

Задача о потоке с фиксированными доплатами появляется как подзадача во многих задачах проектирования транспортных и телекоммуникационных сетей, а также в задачах производственного планирования, включающих выбор схем поставок ресурсов производителям, а готовой продукции — потребителям.

Введем два семейства бинарных переменных:

- x_e — *поток* (количество транспортируемого продукта) по дуге $e \in E$;
- $y_e = 1$, если дуга $e \in E$ используется для транспортировки продукта и $y_e = 0$ в противном случае.

В выбранных переменных задача о потоке с фиксированными доплатами формулируется следующим образом:

$$\sum_{e \in E} (f_e y_e + c_e x_e) \to \min, \tag{4.16a}$$

$$\sum_{e \in E(V,v)} x_e - \sum_{e \in E(v,V)} x_e = d_v, \quad v \in V, \tag{4.16b}$$

$$0 \le x_e \le u_e y_e, \quad e \in E, \tag{4.16c}$$

$$y_e \in \{0, 1\}, \quad e \in E. \tag{4.16d}$$

Здесь для $S, T \subseteq V$ мы обозначаем через $E(S, T)$ множество дуг, выходящих из S и входящих в T.

Целью (4.16a) в этой задаче является минимизация транспортных расходов. *Балансовые уравнения* (4.16b) требуют, чтобы в каждый узел поступало ровно столько потока, сколько требуется. *Переменные верхние границы* (4.16c) задают *ограничения на пропускные способности*, выполнение которых означает, что величина потока по каждой дуге не может превышать пропускной способности этой дуги, и если какая-то дуга не используется для транспортировки продукта ($y_e = 0$), то поток по ней должен быть равен нулю ($x_e = 0$).

4.6.2. Размещение центров обслуживания

Для обслуживания n клиентов отобраны m возможных мест (пунктов) для размещения не более q ($1 \le q \le m$) центров обслуживания (предприятий, складов, станций скорой помощи и т. д.). Для каждого пункта $i = 1, \ldots, m$ задана фиксированная стоимость f_i размещения центра обслуживания и его емкость (сколько клиентов он может обслужить) b_i. Известна также стоимость c_{ij} обслуживания клиента j из пункта i, $j = 1, \ldots, n$, $i = 1, \ldots, m$. Нужно выбрать места для размещения центров обслуживания и прикрепить клиентов к центрам обслуживания таким образом, чтобы минимизировать общую стоимость размещения центров и обслуживания клиентов.

Введем два семейства бинарных переменных:

- $y_i = 1$, если центр размещается в пункте i, и $y_i = 0$ в противном случае;
- $x_{ij} = 1$, если потребитель j обслуживается из пункта i, и $x_{ij} = 0$ в противном случае.

Стандартная формулировка задачи следующая:

$$\sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij} + \sum_{i=1}^{m} f_i y_i \to \min, \tag{4.17a}$$

$$\sum_{i=1}^{n} y_i \le q, \tag{4.17b}$$

$$\sum_{i=1}^{m} x_{ij} = 1, \quad j = 1,\dots,n, \tag{4.17c}$$

$$\sum_{j=1}^{n} x_{ij} \le b_i y_i, \quad i = 1,\dots,m, \tag{4.17d}$$

$$x_{ij} \le y_i, \quad i = 1,\dots,m;\ j = 1,\dots,n, \tag{4.17e}$$

$$y_i \in \{0,1\}, \quad i = 1,\dots,m, \tag{4.17f}$$

$$x_{ij} \in \{0,1\}, \quad i = 1,\dots,m;\ j = 1,\dots,n. \tag{4.17g}$$

Здесь неравенство (4.17b) позволяет разместить не более q центров. Ограничения (4.17c) присоединяют каждого клиента к единственному пункту обслуживания. Из неравенств (4.17d) следует, что не более b_i клиентов могут обслуживаться в пункте i, где размещен обслуживающий центр.

Ограничения (4.17e) призваны усилить данную формулировку. Практика показала, что без ограничений (4.17e) формулировка (4.17) слабая. Известны примеры, когда хорошие коммерческие программы не могли решить задачу ЦП (4.17) без ограничений (4.17e). В то же время, после добавления неравенств (4.17e), те же задачи решались в течение нескольких минут.

Часто величины c_{ij} представляют собой транспортные расходы. Если $q = n$ и не брать в расчет фиксированные затраты на размещение объектов, то оптимальным решением было бы размещение центров во всех возможных местах. С другой стороны, если не учитывать затраты на транспортировку и предположить, что все $b_i = n$, то оптимальное решение состояло бы в том, чтобы разместить только один центр в пункте, где фиксированные затраты минимальны. Таким образом, можно считать, что суть задачи размещения центров обслуживания в том, чтобы оптимально соотнести фиксированные и транспортные расходы.

4.6.3. Размер партии: однопродуктовая модель

Рассмотрим однопродуктовый вариант задачи о размере партии с неограниченными (по сравнению с требованиями) производственными возможностями. Плановый горизонт состоит из T периодов. Для каждого периода $t = 1, \dots, T$ заданы:

- d_t — потребность в некотором продукте;
- f_t — фиксированная стоимость организации производства;
- c_t — стоимость производства единицы продукта;
- h_t — стоимость хранения единицы продукта.

Запасы продукта на складе перед началом планового горизонта равны s_0.

Нужно определить, сколько единиц продукта производить в каждом из периодов, чтобы полностью удовлетворить спрос и суммарные затраты на производство и хранение продукта были минимальны.

Для $t = 1, \dots, T$ введем переменные:

- x_t — количество произведенного продукта за период t;
- s_t — количество продукта, хранимого на складе в конце периода t;
- $y_t = 1$, если в период t организуется производство продукта, и $y_t = 0$ в противном случае.

Определив $D_t = \sum_{\tau=t}^{T} d_t$, мы можем записать следующую формулировку:

$$\sum_{t=1}^{T} (f_t y_t + c_t x_t + h_t s_t) \to \min \tag{4.18a}$$

$$s_{t-1} + x_t = d_t + s_t, \quad t = 1, \dots, T, \tag{4.18b}$$

$$0 \le x_t \le D_t y_t, \quad t = 1, \dots, T, \tag{4.18c}$$

$$y_t \in \{0, 1\}, \quad t = 1, \dots, T. \tag{4.18d}$$

Целевая функция (4.18a) подсчитывает суммарные затраты по всем T периодам, и мы хотим эти затраты минимизировать. Балансовое равенство (4.18b) связывает два соседних периода: то, что было на складе в конце периода $t-1$, плюс произведенное в период t равняется спросу плюс то, что будет храниться на складе в конце периода t. Правая часть неравенства (4.18c) выражает импликацию $y_t = 0 \Rightarrow x_t = 0$.

Если все фиксированные стоимости f_t положительны, то для оптимального решения (x^*, y^*) релаксационной задачи ЛП для задачи (4.18) должны выполняться равенства

$$y^*_t = x^*_t / D_t, \quad t = 1, \dots, T.$$

Следовательно, для всех периодов t, в которых организуется производство ($x^*_t > 0$), за исключением последнего из таких периодов, значения y^*_t дробные, поскольку $y^*_t = x^*_t/D_t \le d_t/D_t < 1$. Много дробных значений для целочисленных компонент у оптимального решения релаксационной задачи ЛП — верный признак того, что используемая формулировка слабая.

Чтобы получить идеальную формулировку, к системе ограничений задачи (4.18) нужно добавить систему (l, S)-неравенств:

$$\sum_{t \in S} x_t + \sum_{t \in \bar{S}} d_{tl}\, y_t \ge d_{1l}, \quad S \subseteq \{1, \dots, l\},\ 1 \le l \le T. \tag{4.19}$$

Здесь $\bar{S} = \{1, \dots, l\} \setminus S$ и $d_{ij} = \sum_{t=i}^{j} d_t$. Неравенства (4.19) выражают следующее простое наблюдение: сумма произведенного в периоды $t \in S$ ($\sum_{t\in S} x_t$) и максимума из того, что можно произвести в периоды $t \in \bar{S}$ для использования в первые l периодов ($\sum_{t \in \bar{S}} d_{tl}\, y_t$), должна быть не меньше потребности в продукте (d_{1l}) в эти первые l периодов.

Мы видим, что идеальная формулировка для множества допустимых решений задачи (4.18) содержит экспоненциально большое число неравенств. Мы можем усилить формулировку (4.18) и другим способом, *дизагрегировав переменные* x_t: $x_t = = \sum_{\tau=t}^{T} x_{t\tau}$, где новая переменная $x_{t\tau}$ представляет количество продукта, произведенного в период $t = 1, \dots, T$ для удовлетворения потребности в период $\tau = t, \dots, T$.

Сначала исключим из формулировки переменные s_t. Сложив балансовые равенства $s_{k-1} + x_k = d_k + s_k$ для $k = 1, \dots, t$, получим

$$s_t = \sum_{k=1}^{t} x_k - \sum_{k=1}^{t} d_k.$$

Используя эти равенства, мы перепишем целевую функцию (4.18a) следующим образом

$$\sum_{t=1}^{T}(f_t y_t + c_t x_t + h_t\left(\sum_{k=1}^{t} x_k - \sum_{k=1}^{t} d_k\right) =$$
$$= \sum_{t=1}^{T}(f_t y_t + w_t x_t) - K = \sum_{t=1}^{T} f_t y_t + \sum_{t=1}^{T}\sum_{\tau=t}^{T} w_t x_{t\tau} - K,$$

где $w_t = c_t + h_t + \cdots + h_T$ и $K = \sum_{t=1}^{T} h_t \left(\sum_{k=1}^{t} d_k\right)$.

В новых переменных $x_{t\tau}$ задачу (4.18) можно переформулировать следующим образом:

$$\begin{aligned} &\sum_{t=1}^{T} f_t y_t + \sum_{t=1}^{T}\sum_{\tau=t}^{T} w_t x_{t\tau} \to \min,\\ &\sum_{t=1}^{\tau} x_{t\tau} = d_\tau, \quad \tau = 1,\dots,T,\\ &0 \le x_{t\tau} \le d_\tau y_t, \quad t = 1,\dots,T;\ \tau = t,\dots,T,\\ &y_t \in \{0,1\}, \quad t = 1,\dots,T. \end{aligned} \tag{4.20}$$

Нетрудно показать, что среди решений релаксационной задачи ЛП для задачи (4.20) есть такие решения (x^*, y^*), для которых все компоненты вектора y^* целочисленны и из $x^*_{t\tau} > 0$ следует, что $x^*_{t\tau} = d_\tau$, т. е. весь продукт, потребляемый в период τ, полностью производится только в одном периоде. По этой причине формулировку (4.20) можно считать «почти» идеальной.

Основной недостаток многих расширенных формулировок — это их большой размер. В нашем случае мы заменили формулировку (4.18) с $3T$ переменными и $2T$ нетривиальными[7] ограничениями на формулировку (4.20) с $T(T+1)/2$ переменными и $2T$ ограничениями. Для примера, при $T = 100$ в первом случае мы имеем только 300 переменных, а во втором — 5050. Разница огромная! На практике иногда эффективнее использовать так называемую *приближенную расширенную формулировку*, которая получается присоединением к основной компактной формулировке части «самых важных» переменных и ограничений расширенной формулировки.

[7] Обычно тривиальными ограничениями называют нижние и верхние границы на переменные.

4.6.4. Размер партии: многопродуктовая модель

Нужно определить план производства n различных продуктов на m машинах в течение временного горизонта, разделенного на T периодов. Пусть M_j обозначает множество машин, способных производить продукт j. Исходные данные:

- f_{it} — фиксированная стоимость организации производства на машине i в период t;
- c_{ijt} — стоимость производства единицы продукта j на машине i в период t;
- h_{jt} — стоимость хранения единицы продукта j в период t;
- d_{jt} — потребность в продукте j в период t;
- $T_{it}^{\min}, T_{it}^{\max}$ — минимальное и максимальное время работы машины i в период t;
- ρ_{ijk} — количество единиц продукта j, используемого для производства единицы продукта k на машине $i \in M_k$;
- τ_{ij} — время производства единицы продукта j на машине $i \in M_j$;
- s_{j0} — запас продукта j перед началом планового горизонта.

Нужно определить, сколько производить каждого из продуктов и в какие периоды, чтобы удовлетворить потребности при минимальных суммарных затратах на производство и хранение продукции.

Введем переменные:

- x_{ijt} — количество продукта j, производимого в период t на машине i;
- s_{jt} — количество продукта j, хранимого на складе в конце периода t;
- $y_{it} = 1$, если машина i работает в период t, и $y_{it} = 0$ в противном случае.

Теперь мы формулируем следующую задачу:

$$\sum_{t=1}^{T}\sum_{j=1}^{n}\left(h_{jt}\, s_{jt} + \sum_{i\in M_j} c_{ijt}\, x_{ijt}\right) + \sum_{t=1}^{T}\sum_{i=1}^{m} f_{it}\, y_{it} \to \min, \tag{4.21a}$$

$$s_{j,t-1} + \sum_{i\in M_j} x_{ijt} = d_{jt} + s_{jt} + \sum_{k=1}^{n}\sum_{i\in M_k} \rho_{ijk}\, x_{ikt},$$
$$j = 1,\dots,n;\ t = 1,\dots,T, \tag{4.21b}$$

$$T_{it}^{\min} y_{it} \le \sum_{j:i\in M_j} \tau_{ij}\, x_{ijt} \le T_{it}^{\max} y_{it}, \quad i=1,\dots,m;\ t=1,\dots,T, \tag{4.21c}$$

$$s_{jt} \ge 0, \quad j=1,\dots,n;\ t=1,\dots,T, \tag{4.21d}$$

$$x_{ijt} \ge 0,\ y_{ijt} \in \{0,1\}, \quad j=1,\dots,n;\ i\in M_j\,;\ t=1,\dots,T. \tag{4.21e}$$

Целевая функция (4.21a) предписывает минимизировать суммарные издержки производства. Балансовые ограничения (4.21b) обеспечивают переход из одного периода в следующий: количество продукта на складе в период $t-1$ плюс то, что произведено в период t, должно равняться потребности (проданному) в период t плюс то, что используется для производства других продуктов, а также то, что будет храниться на складе в течение следующего периода. Неравенства (4.21c) требуют, чтобы время работы машин в каждом из периодов было в пределах заданных лимитов, причем если машина i не работает в период t, то она не может ничего производить (все x_{ijt} равны нулю).

4.6.5. Планирование производства электроэнергии

Планирование производства электроэнергии предполагает разработку получасового (или часового) расписания на неделю (или на сутки) для каждого генератора, т. е. нужно указать, когда генераторы должны работать и сколько энергии производить. В типичной энергосистеме имеются разные типы генераторов. На одном полюсе — ядерные энергоблоки, которые производят электроэнергию с очень малым увеличением стоимости на каждый дополнительный мегаватт/час, но при этом после остановки вновь запустить такой энергоблок достаточно дорого и к тому же требует много времени. Обычно ядерные энергоблоки останавливают в периоды наименьшего потребления энергии (например, летом, когда энергия не тратится на отопление и спрос наименьший). На другом полюсе — газовые турбогенераторы, которые могут начать производить электроэнергию в считанные минуты, но с большим увеличением стоимости на каждый дополнительный мегаватт/час.

В нормальной ситуации стандартная политика управления генераторами состоит в том, чтобы при росте потребления сначала запускать генераторы, которые наиболее эффективны, но имеют большую стоимость запуска, а в самом конце запускать генераторы, которые наименее эффективны, но имеют низкую стоимость запуска. Но при возникновении краткосрочных пиков потребления (скажем, во время телетрансляций важных футбольных матчей) может оказаться наиболее экономичным в первую очередь быстро запустить не самые эффективные, но менее инертные генераторы. Задача экономичного управления генераторами усложняется, когда в энергосистеме также работают генераторы, характеристики которых не столь полярны.

Перейдем к формальному описанию модели. Пусть T есть число периодов в плановом горизонте. Считаем, что период 1 циклически следует за периодом T. Мы знаем *потребность* в энергии d_t для каждого периода t. Для того чтобы всегда быть готовым быстро увеличить производство энергии в случае непредвиденного увеличения потребления, в любой период мощность *активных* (работающих в данный период) генераторов должна быть не менее чем в q раз больше потребности.

Пусть имеется n генераторов, а i-й генератор имеет следующие характеристики:

- l_i, u_i — минимальная и максимальная рабочая мощность;
- r_i^-, r_i^+ — параметры инерционности (мощность работающего генератора в следующем периоде не может уменьшиться (соответственно увеличиться) более чем на r_i^- (соответственно r_i^+);
- g_i — стоимость запуска неработающего генератора;
- f_i, p_i — фиксированная и удельная стоимости (если в некоторый период генератор работает на мощности v, то это стоит $f_i + p_i v$).

Естественным является следующий выбор переменных:

- $x_{it} = 1$, если i-й генератор работает в период t, и $x_{it} = 0$ в противном случае;
- $z_{it} = 1$, если i-й генератор включается в период t, и $z_{it} = 0$ в противном случае;
- y_{it} — количество электроэнергии, производимой генератором i в период t.

Теперь мы можем записать типичную модель для рассматриваемой задачи:

$$\sum_{i=1}^{n} \sum_{t=1}^{T} (g_i z_{it} + f_i x_{it} + p_i y_{it}) \to \min, \tag{4.22a}$$

$$\sum_{i=1}^{n} y_{it} = d_t, \quad t = 1, \dots, T, \tag{4.22b}$$

$$\sum_{i=1}^{n} u_i x_{it} \geq q\, d_t, \quad t = 1, \ldots, T, \tag{4.22c}$$

$$l_i x_{it} \leq y_{it} \leq u_i x_{it}, \quad i = 1, \ldots, n;\ t = 1, \ldots, T, \tag{4.22d}$$

$$-r_i^- \leq y_{i1} - y_{i,T} \leq r_i^+, \quad i = 1, \ldots, n, \tag{4.22e}$$

$$-r_i^- \leq y_{it} - y_{i,t-1} \leq r_i^+, \quad i = 1, \ldots, n;\ t = 2, \ldots, T, \tag{4.22f}$$

$$x_{i1} - x_{i,T} \leq z_{it}, \quad i = 1, \ldots, n, \tag{4.22g}$$

$$x_{it} - x_{i,t-1} \leq z_{it}, \quad i = 1, \ldots, n;\ t = 2, \ldots, T, \tag{4.22h}$$

$$x_{it}, z_{it} \in \{0, 1\}, \quad i = 1, \ldots, n;\ t = 1, \ldots, T, \tag{4.22i}$$

$$y_{it} \geq 0, \quad i = 1, \ldots, n;\ t = 1, \ldots, T. \tag{4.22j}$$

Целью (4.22a) в данной задаче является минимизация затрат при производстве электроэнергии. Равенства (4.22b) требуют, чтобы производилось ровно столько энергии, сколько требуется. Ограничения (4.22c) обеспечивают «устойчивость» расписания: в любой период времени суммарная максимальная мощность всех работающих генераторов должна не менее чем в q раз превосходить потребность в данный период. Неравенства (4.22d) гарантируют, что количество электроэнергии, производимой каждым из генераторов, должно быть в пределах между минимальной и максимальной мощностями этого генератора. Неравенства (4.22e) и (4.22f) ограничивают изменения мощностей генераторов за один период. И наконец, неравенства (4.22g) и (4.22h) выражают логическую зависимость между переменными x и z: генератор включается в период t ($z_{it} = 1$) только в том случае, когда он не работал в предыдущий период ($x_{i,t-1} = 0$) и работает в текущий период ($x_{it} = 1$). Если $g_i = 0$, то значение переменной z_{it} не влияет на целевую функцию и может быть произвольным. Если $g_i > 0$, то z_{it} примет значение 1 только в том случае, когда $x_{it} - x_{i,t-1} = 1$, т. е. только тогда, когда генератор включается.

4.7. Упражнения

4.1. Фермер, который обрабатывает 100 га. земли, хочет составить производственный план на ближайшие пять лет.

В настоящий момент фермер имеет стадо из 120 коров, 100 из которых — дойные коровы, а 20 — телки. Предположим, что в стаде ровно по 10 коров всех возрастов от 0 (новорожденные телки) до 11 лет.

В год для содержания одной дойной коровы требуется обрабатывать

1 га земли, а для содержания одной телку — 2/3 га. В среднем, одна корова рожает в год одного теленка. Половина из всех рожденных телят — бычки, которых продают почти сразу после рождения за \$50 за бычка. Оставшаяся половина — телки, каждую из которых можно продать за \$60, или оставить в стаде, чтобы через два года вырастить дойную корову. Как только дойная корова достигает возраста 12 лет, фермер продает ее в среднем за \$180. Годовые потери в стаде составляют 5 % дойных коров и 3 % телок.

В год одна корова дает молока на \$600. В настоящий момент в коровнике можно разместить не более 125 коров. Для размещения одной дополнительной коровы необходимо вложить \$300. В год каждая дойная корова потребляет 0.7 тонны зерна и 0.8 тонны кормовой свеклы. Зерно и свеклу можно выращивать на ферме. Каждый гектар земли дает в среднем 0.6 тонны зерна и 1.5 тонны свеклы. Зерно можно купить по цене \$150 и продать по цене \$125 за тонну. Соответственно, тонну свеклы можно купить за \$100 и продать за \$80.

Потребности в трудовых ресурсах (часов в год) следующие:

Дойная корова	Телка	1 га. зерновых	1 га. свеклы
15	60	10	30

Иные издержки (электроэнергия, топливо, ветеринарное обслуживание, удобрения и т. д.) следующие:

Дойная корова	Телка	1 га. зерновых	1 га. свеклы
\$150	\$75	\$50	\$40

На ферме работают 3 постоянных рабочих, каждый из которых может выполнять любую работу. Заработная плата каждого рабочего \$6000 в год, и в среднем рабочий за один год отрабатывае 2000 часов. Имется возможность нанимать временных рабочих с оплатой \$1.5 за час работы.

В настоящий момент фемер может получить заем (размер которого нужно определить) на 10 лет под 15 % годовых. Возврат заемных средст и процентов по ним должен осуществлять каждый год 10-ю равными частями. По окончании пятилетнего срока фермер допускает изменение численности дойных коров не более чем на 25 % от настоящего количества (100 коров).

Сформулируйте задачу СЦП, решив которую фермер определит пятилетний план работы, который за пять лет обеспечит ему наибольшую прибыль.

4.2. Решите следующие примеры сначала методом сечений, а затем методом ветвей и границ:

а)
$$\begin{aligned} x_1 + x_2 &\to \max, \\ 2x_1 + x_2 &\le 5, \\ x_1 + 2x_2 &\le 3, \\ x_1, x_2 &\in \mathbb{Z}_+, \end{aligned}$$

б)
$$\begin{aligned} x_1 + x_2 &\to \max, \\ 3x_1 + x_2 &\le 6, \\ x_1 + 2x_2 &\le 6, \\ x_1, x_2 &\in \mathbb{Z}_+. \end{aligned}$$

Глава 5
Динамическое программирование

Метод *динамического программирования*, как и метод ветвей и границ, при поиске оптимального решения некоторой задачи производит разумный перебор допустимых решений, но делает это другим способом. Вычисления в конкретном методе динамического программирования проводятся по рекуррентной формуле, которая сводит решение задачи к решению нескольких задач того же самого типа, но уже меньшего размера.

Применения метода динамического программирования разнообразны, что не позволяет сформулировать общую задачу, обобщающую все задачи, решаемые этим методом. Тем не менее, мы начинаем эту главу с рассмотрения достаточно общего процесса принятия решений, оптимизация которого проводится методом динамического программирования. В последующих разделах изучаются алгоритмы динамического программирования для решения конкретных оптимизационных задач. В заключительном разделе данной главы общая схема динамического программирования, представленная в первом разделе главы, расширяется для решения задач с неопределенными параметрами.

5.1. Общая схема

Рассмотрим процесс принятия решений, состоящий из m стадий: вначале система находится в состоянии s_0, если после стадии $i-1$ ($i = 1,\dots,m$) система находится в состояние s_{i-1}, а на стадии i принято решение $x^i \in D_i(s_{i-1})$, то мы получаем *доход*

$$g_i(x_i, s_{i-1}), \tag{5.1}$$

а система перейдет в состояние

$$s_i = tr_i(x_i, s_{i-1}) \in S_i, \tag{5.2}$$

где $D_i(s_{i-1})$ обозначает множество допустимых решений на стадии i в состоянии s_{i-1}, а S_i — множество возможных состояний после стадии i, $S_0 = \{s_0\}$.

Наша цель — максимизировать суммарный (по всем стадиям) доход:

$$\begin{gathered}\max \sum_{i=1}^{m} g_i(x_i, s_{i-1}),\\ s_i = tr_i(x_i, s_{i-1}),\ x_i \in D_i(s_{i-1}), \quad i = 1, \ldots, m.\end{gathered} \tag{5.3}$$

5.1.1. Прямая индукция

Для $k = 1, \ldots, m$ определим оптимальный суммарный доход для процесса из k первых стадий и начальным состоянием s_0 и конечным состоянием $s_k \in S_k$:

$$\begin{aligned}V_k(s_k) \stackrel{\text{def}}{=} \max\Biggl\{\sum_{i=1}^{k} g_i(x_i, s_{i-1}) : s_i = tr_i(x_i, s_{i-1}),\ x_i \in D_i(s_{i-1}),\\ s_{i-1} \in S_{i-1}, \quad i = 1, \ldots, k\Biggr\}.\end{aligned} \tag{5.4}$$

Как обычно, предполагается, что максимум, вычисленный по пустому множеству альтернатив, равен $-\infty$.

Наша цель — максимизировать суммарный доход процесса из m стадий:

$$\max_{s_m \in S_m} \max_{x_1, \ldots, x_m} V_m(s_m).$$

Так как

$$\begin{aligned}V_k(s_k) = \max\Biggl\{\max\Biggl\{\sum_{i=1}^{k-1} g_i(x_i, s_{i-1}) :\ s_i = tr_i(x_i, s_{i-1}),\ x_i \in D_i(s_{i-1}),\\ s_{i-1} \in S_{i-1},\ i = 1, \ldots, k-1\Biggr\} +\\ g_k(x_k, s_{k-1}) :\ s_k = tr_k(x_k, s_{k-1}),\ x_k \in D_k(s_{k-1}),\ s_{k-1} \in S_{k-1}\Biggr\},\end{aligned}$$

то мы получаем следующую рекуррентную формулу:

$$V_k(s_k) = \max\{V_{k-1}(s_{k-1}) + g_k(x_k, s_{k-1}) : \ s_k = tr_k(x_k, s_{k-1}), \\ x_k \in D_k(s_{k-1}), \ s_{k-1} \in S_{k-1}\}, \quad k = 1, \dots, m. \tag{5.5}$$

В (5.5) предполагается, что $v_0(s_0) \stackrel{\text{def}}{=} 0$.

5.1.2. Обратная индукция

Для $k = 1, \dots, m$ определим оптимальный суммарный доход для процесса из $m-k+1$ последних стадий и начальным состоянием $s_{k-1} \in S_{k-1}$:

$$U_k(s_{k-1}) \stackrel{\text{def}}{=} \max \sum_{i=k}^{m} g_i(x_i, s_{i-1}), \\ s_i = tr_i(x_i, s_{i-1}), \ x_i \in D_i(s_{i-1}), \ s_i \in S_i, \quad i = k, \dots, m. \tag{5.6}$$

Наша цель — максимизировать суммарный доход процесса из m стадий и начальным состоянием s_0:

$$\max_{x_1, \dots, x_m} u_1(s_0).$$

Справедлива следующая рекуррентную формулу:

$$U_k(s_{k-1}) = \max\{U_{k+1}(s_k) + g_k(x_k, s_{k-1}) : \\ s_k = tr_k(x_k, s_{k-1}), \ x_k \in D_k(s_{k-1}, \ s_k \in S_k)\}, \quad k = m, \dots, 1. \tag{5.7}$$

В (5.7) предполагается, что $U_{m+1}(s_m) \stackrel{\text{def}}{=} 0$ для всех состояний $s_m \in S_m$, где S_m обозначает множество всех возможных состояний после стадии m.

5.2. Задача о рюкзаке

Имеется два основных варианта *задачи о рюкзаке*:

целочисленный рюкзак:

$$\max\{c^T x : \ a^T x \le b, \ x \in \mathbb{Z}^n_+\}, \tag{5.8}$$

0,1-рюкзак:

$$\max\{c^T x : \ a^T x \leq b, \ x \in \{0,1\}^n\}. \tag{5.9}$$

Здесь $c \in \mathbb{R}^n_{++}$, $a \in \mathbb{Z}^n_{++}$, b — положительное целое число.

Задача получила свое название из-за следующей шутливой интерпретации. Вы выиграли приз, который позволяет вам в супермаркете наполнить ваш рюкзак любыми имеющимися там предметами, при условии, что суммарный вес предметов не должен превосходить заданной величины b. Предположим, что в супермаркете имеется n типов предметов, а стоимость одного предмета типа j равна c_j. Вы, скорее всего, захотите наполнить свой рюкзак таким образом, чтобы суммарная стоимость предметов в нем была максимальной. Для этого вам придется решить задачу о целочисленном рюкзаке. Если разрешается взять не более одного экземпляра каждого из предметов, то тогда вам нужно решать задачу о 0,1-рюкзаке.

Разумеется, мы могли бы привести и более серьезные примеры применения задачи о рюкзаке на практике. Но скажем сразу, реальный мир не так прост, и в нем найдется не много реальных ситуаций, которые можно смоделировать только одним ограничением пусть и с целочисленными переменными[8]. Мы здесь изучаем задачу о рюкзаке по другой причине: задачи об отделении для многих классов неравенств, а также задачи оценивания для ряда алгоритмов с генерацией столбцов формулируются как задача о рюкзаке.

Задачи о рюкзаке — это самые простые задачи ЦП, но даже они (как (5.8), так и (5.9)) являются **NP**-трудными[9]. Несмотря на это, на практике мы можем решать достаточно большие задачи о рюкзаке сравнительно быстро методом динамического программирования.

5.2.1. 0,1-рюкзак

Теперь рассмотрим задачу (5.9) о 0,1-рюкзаке. Представим метод решения этой задачи в виде процесса из $m = n$ стадий: на стадии мы

[8] Иногда, возражая против этого утверждения, вспоминают агрегацию уравнений (см. упр. 5.2), которая позволяет выразить несколько ограничений всего лишь одним уравнением. Но агрегация всегда ухудшает формулировку, и на практике ее следует избегать.

[9] В теории сложности вычислений через **NP** обозначают класс задач распознавания (с ответом «да» или «нет»), разрешимых на недетерминированной машине Тьюринга за полиномиальное время. Задача $\mathcal{P}$ называется **NP**-трудной, если любую задачу из класса **NP** можно решить за полиномиальное время, используя процедуру для решения задачи $\mathcal{P}$, при этом время работы такой процедуры считается равным 1.

принимаем решение $x_i \in \{0,1\}$, после чего система перейдет в состояние $s_i = \sum_{j=1}^{i} a_j x_j$. Начальное состояние $s_0 = 0$. Для $i = 1,\dots,n$ введем обозначения:

$$\begin{aligned}
S_i &\stackrel{\text{def}}{=} \{0,1,\dots,b\},\\
D_i(s_{i-1}) &\stackrel{\text{def}}{=} \{x_i \in \{0,1\} : s_{i-1} + a_i \le b\}, \quad s_{i-1} \in S_{i-1},\\
g_i(x_i, s_{i-1}) &\stackrel{\text{def}}{=} c_i x_i, \quad tr_i(x_i, s_{i-1}) \stackrel{\text{def}}{=} s_{i-1} + a_i x_i, \quad x_i \in \{0,1\}, s_{i-1} \in S_{i-1},
\end{aligned}$$

Если обозначить s_k через β, с учетом того, что $s_{k-1} = \beta - a_k x_k$, рекуррентная формула (5.5) примет следующий вид:

$$V_k(\beta) = \max\{V_{k-1}(\beta - a_k x_k) + c_k x_k :\ x_k \in \{0,1\},\ a_k x_k \le \beta\}, \quad k = 1,\dots,n. \tag{5.10}$$

Исключая из (5.10) x_k, для $k = 1,\dots,n$ получим следующее выражение:

$$V_k(\beta) = \begin{cases} V_{k-1}(\beta), & \beta = 0,\dots,a_k - 1,\\ \max\{V_{k-1}(\beta), V_{k-1}(\beta - a_k) + c_k\}, & \beta = a_k,\dots,b, \end{cases} \tag{5.11}$$

при начальных условиях

$$\begin{aligned}
V_0(0) &= 0,\\
V_0(\beta) &= -\infty, \quad \beta = 1,\dots,b.
\end{aligned}$$

Для $\beta = 1,\dots,b$ равенство $V_0(\beta) = -\infty$ можно интерпретировать как выражение того факта, что $\beta \notin S_0$.

Поскольку нам нужно вычислить $n \cdot b$ значений $V_k(\beta)$ и при вычислении каждого из этих значений мы выполняем одно сравнение и одно присваивание, то вычисления по рекуррентной формуле (5.11) можно выполнить за время $O(n \cdot b)$, используя $O(n \cdot b)$ ячеек памяти.

По определению величин $V_k(\beta)$, оптимальное значение целевой функции в задаче (5.9) равно $\max\limits_{0 \le \beta \le b} V_n(\beta)$. Вычислив все значения $V_k(\beta)$, оптимальное решение x^* можно найти, выполнив следующий *обратный ход*.

Начинаем с $\beta \in \arg\max\limits_{0 \le q \le b} V_n(q)$. Для $k = n,\dots,1$ положить $x_k^* = 0$, если $V_k(\beta) = V_{k-1}(\beta)$, а если $V_k(\beta) = V_{k-1}(\beta - a_k) + c_k$, то положить $x_k^* = 1$ и $\beta := \beta - a_k$.

Формула (5.11) не единственно возможная. Новую формулу мы могли бы получить также исходя из общей рекуррентной формулы (5.5). Но эту формулу все же проще получить как аналог формулы (5.11).

Предположим, что все стоимости c_j целочисленны. Пусть $C \in \mathbb{Z}$ есть оценка сверху для оптимального значения целевой функции в задаче (5.9). Для $k = 1, \ldots, n$ и $z = 0, \ldots, C$ определим

$$G_k(z) \stackrel{\text{def}}{=} \min \left\{ \sum_{j=1}^{k} a_j x_j \ : \ \sum_{j=1}^{k} c_j x_j = z,\ x_j \in \{0,1\},\ j = 1, \ldots, k \right\}.$$

Для $k = 1, \ldots, n$ справедлива следующая рекуррентная формула:

$$G_k(z) = \begin{cases} G_{k-1}(z), & z = 0, \ldots, c_k - 1, \\ \min\{G_{k-1}(z), G_{k-1}(z - c_k) + a_k\}, & z = c_k, \ldots, C, \end{cases} \tag{5.12}$$

при начальных условиях

$$G_0(0) = 0,$$
$$G_0(z) = \infty, \quad z \neq 0.$$

Оптимальное значение целевой функции в задаче (5.9) равно

$$\max\{z : G_n(z) \leq b\}.$$

Вычислив все значения $G_k(z)$, оптимальное решение x^* можно найти, выполнив следующий *обратный ход*.

Начинаем с $z \in \arg\max\{q : G_n(q) \leq b\}$. Для $k = n, \ldots, 1$, если $G_k(z) = G_{k-1}(z)$, положить $x_k^* = 0$, а, если $G_k(z) = G_{k-1}(z - c_k) + a_k$, положить $x_k^* = 1$ и $z := z - c_k$.

Вычисления по формуле (5.12) можно выполнить за время $O(n \cdot C)$, используя $O(n \cdot C)$ ячеек памяти.

Сравнивая сложности вычисления по обеим рекуррентным формулам, можно прийти к простому заключению: формулу (5.11) нужно использовать, если $b < C$, иначе нужно использовать формулу (5.12).

Пример 5.1. *Решим задачу*

$$\begin{aligned} 10x_1 + 7x_2 + 25x_3 + 24x_4 &\to \max, \\ 2x_1 + 1x_2 + 6x_3 + 5x_4 &\leq 7, \\ x_1, x_2, x_3, x_4 &\in \{0,1\}. \end{aligned}$$

Решение. Очевидно, что оптимальное значение целевой функции в этой задаче больше $b = 7$. Поэтому мы используем рекуррентную формулу (5.11). Вычисления представлены в табл. 5.1. Оптимальное значе-

Таблица 5.1
Вычисления для примера 5.1

β	V_0	V_1	V_2	V_3	V_4
0	0	0	0	0	0
1	$-\infty$	$-\infty$	7	7	7
2	$-\infty$	10	10	10	10
3	$-\infty$	$-\infty$	17	17	17
4	$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$
5	$-\infty$	$-\infty$	$-\infty$	$-\infty$	24
6	$-\infty$	$-\infty$	$-\infty$	25	31
7	$-\infty$	$-\infty$	$-\infty$	32	34

ние целевой функции для нашего примера равно

$$\max_{0 \le q \le 7} V_4(q) = V_4(7) = 34.$$

Чтобы найти само оптимальное решение x^*, нужно выполнить обратный ход по формуле (5.11), начиная с $\beta = 7$:

$$\begin{aligned}
V_4(7) &= \max\{V_3(7), F_3(7-5) + 24\} = \max\{32, 10 + 24\} = 34 \Rightarrow \\
&\Rightarrow x_4^* = 1 \text{ и } \beta = 7 - 5 = 2; \\
V_3(2) &= V_2(2) = 10 \Rightarrow \\
&\Rightarrow x_3^* = 0 \text{ и } \beta = 2; \\
V_2(2) &= \max\{V_1(2), V_1(2-1) + 7\} = \max\{10, 0 + 7\} = 10 \Rightarrow \\
&\Rightarrow x_2^* = 0 \text{ и } \beta = 2; \\
V_1(2) &= \max\{V_0(2), F_0(2-2) + 10\} = \max\{-\infty, 0 + 10\} = 10 \Rightarrow \\
&\Rightarrow x_1^* = 1 \text{ и } \beta = 0.
\end{aligned}$$

Значит, точка $x^* = (1, 0, 0, 1)^T$ есть оптимальное решение для этого примера. $\square$

Пример 5.2. *Решим задачу*

$$\begin{aligned}
2x_1 + \quad x_2 + \quad 3x_3 + \quad 4x_4 &\to \max, \\
35x_1 + 24x_2 + 69x_3 + 75x_4 &\le 100, \\
x_1, x_2, x_3, x_4 &\in \{0, 1\}.
\end{aligned}$$

Таблица 5.2

Вычисления для примера 5.2

z	G_0	G_1	G_2	G_3	G_4
0	0	0	0	0	0
1	∞	∞	24	24	24
2	∞	35	35	35	35
3	∞	∞	59	59	59
4	∞	∞	∞	93	75
5	∞	∞	∞	104	99

Решение. Сначала оценим сверху оптимальное значение целевой функции. Для этого решим *релаксационную задачу ЛП*, которая получается из исходной задачи, отбрасыванием условия целочисленности переменных, позволяя им принимать значения из интервала $[0,1]$. Алгоритм решения задач ЛП с одним ограничением очень прост (см. упр. 2.6).

1. Сортируем отношения c_j/a_j по невозрастанию:

$$\frac{c_1}{a_1} = \frac{2}{35} \geq \frac{c_4}{a_4} = \frac{4}{75} \geq \frac{c_3}{a_3} = \frac{3}{69} \geq \frac{c_2}{a_2} = \frac{1}{24}.$$

2. Вычисляем решение $\hat{x}$:

$$\hat{x}_1 = 1,\ \hat{x}_4 = (100-35)/75 = 13/15,\ \hat{x}_3 = \hat{x}_2 = 0.$$

В качестве верхней оценки возьмем число

$$C = \lfloor c^T\hat{x} \rfloor = \lfloor 2 + 4\cdot(13/15) \rfloor = 5.$$

Поскольку $C = 5 < 100 = b$, то будем использовать формулу (5.12). Вычисления представлены в табл. 5.2.

Оптимальное значение целевой функции для нашего примера равно 5, поскольку $G_4(5) = 99 < 100 = b$. Чтобы найти само оптимальное решение x^*, нужно выполнить обратный ход по формуле (5.12), начиная с $z = 5$:

$$\begin{aligned}
G_4(5) &= \min\{G_3(5), G_3(5-4)+75\} = \min\{104, 24+75\} = 99 \Rightarrow \\
&\Rightarrow x_4^* = 1 \text{ и } z = 5-4 = 1; \\
G_3(1) &= G_2(1) = 24 \Rightarrow \\
&\Rightarrow x_3^* = 0;
\end{aligned}$$

$$
\begin{aligned}
G_2(1) &= \max\{G_1(1), G_1(1-1)+24\} = \max\{\infty, 0+24\} = 24 \Rightarrow \\
&\Rightarrow x_2^* = 1 \text{ и } z = 0; \\
G_1(0) &= G_0(0) = 0 \Rightarrow \\
&\Rightarrow x_1^* = 0.
\end{aligned}
$$

Итак, точка $x^* = (0,1,0,1)^T$ есть оптимальное решение данного примера. □

5.2.2. Целочисленный рюкзак

Мы могли бы представить процесс решения задачи о целочисленном рюкзаке (5.8) в виде процесса из $m = n$ стадий и затем получить рекуррентную формулу. Но эта формула не была бы самой лучшей (с точки зрения эффективности вычислений). Как мы уже отмечали, применения динамического программирования существенно шире применений процесса принятия решений из раздела 5.1. Наиболее эффективную рекуррентную формулу для задачи о целочисленном рюкзаке мы получим иным способом на основании очень простого наблюдения.

Рассмотрим задачу о целочисленном рюкзаке (5.8). Для $\beta = 0,\dots,b$ определим

$$F(\beta) \stackrel{\text{def}}{=} \max\{c^T x:\ a^T x = \beta,\ x \in \mathbb{Z}_+^n\}.$$

Мы можем интерпретировать $F(\beta)$ как стоимость оптимального рюкзака веса β. Если в оптимальном рюкзаке находится хотя бы один предмет j, то $F(\beta) = F(\beta - a_j) + c_j$. Исходя из этого наблюдения, получаем следующую рекуррентную формулу:

$$
\begin{aligned}
&F(0) = 0, \\
&F(\beta) = \max_{j:\ a_j \le \beta} F(\beta - a_j) + c_j, \quad \beta = 1,\dots,b.
\end{aligned}
\tag{5.13}
$$

Как обычно, считаем, что максимум по пустому множеству альтернатив равен $-\infty$.

Вычисления значений $F(\beta)$ по формуле (5.13) называют *прямым ходом* динамического программирования. Когда все значения $F(\beta)$ вычислены, решение x^* задачи (5.8) можно найти, выполнив следующий *обратный ход*.

Начинаем с $\beta \in \arg\max_{0 \le q \le b} F(q)$ и полагаем $x_j^* = 0$ для $j = 1, \dots, n$. Пока $\beta > 0$, выполняем следующие вычисления: находим такой индекс j, что $F(\beta) = F(\beta - a_j) + +c_j$, и полагаем $x_j^* := x_j^* + 1$, $\beta := \beta - a_j$.

Продемонстрируем работу описанного алгоритма на примере.

Пример 5.3. *Решим задачу*

$$\begin{aligned} 4x_1 + 5x_2 + x_3 + 2x_4 &\to \max, \\ 5x_1 + 4x_2 + 2x_3 + 3x_4 &\le 7, \\ x_1, x_2, x_3, x_4 &\in \mathbb{Z}_+. \end{aligned}$$

Решение. Сначала вычисляем

$$\begin{aligned} &F(0) = 0, \\ &F(1) = -\infty, \\ &F(2) = F(0) + 1 = 1, \\ &F(3) = \max\{F(1) + 1, F(0) + 2\} = \max\{-\infty, 2\} = 2, \\ &F(4) = \max\{F(0) + 5, F(2) + 1, F(1) + 2\} = \max\{5, 2, -\infty\} = 5, \\ &F(5) = \max\{F(0) + 4, F(1) + 5, F(3) + 1, F(2) + 2\} = \\ &\qquad = \max\{4, -\infty, 3, 3\} = 4, \\ &F(6) = \max\{F(1) + 4, F(2) + 5, F(4) + 1, F(3) + 2\} = \\ &\qquad = \max\{-\infty, 6, 6, 4\} = 6, \\ &F(7) = \max\{F(2) + 4, F(3) + 5, F(5) + 1, F(4) + 2\} = \max\{5, 7, 5, 7\} = 7. \end{aligned}$$

Теперь найдем оптимальное решение x^*. Поскольку $F(7) = \max_{0 \le q \le 7} F(q)$, то начинаем обратный ход с $\beta = 7$ и $x^* = (0, 0, 0, 0)^T$. Так как $F(7) = F(7 - a_2) + c_2$, то полагаем $x_2^* = 0 + 1 = 1$ и $\beta = 7 - a_2 = 3$. Поскольку $F(3) = F(3 - a_4) + c_4$, то полагаем $x_4^* = 0 + 1 = 1$ и $\beta = 3 - a_4 = 0$. Следовательно, точка $x^* = (0, 1, 0, 1)^T$ — решение нашего примера. □

5.3. Размер партии: однопродуктовая модель

Напомним постановку однопродуктовой задачи о размере партии из раздела 4.6.3. Плановый горизонт состоит из T периодов. Для каждого периода $t = 1, \dots, T$ заданы:

- d_t — потребность в некотором продукте;
- f_t — фиксированная стоимость организации поставки партии продукта;
- c_t — стоимость поставки единицы продукта;
- h_t — стоимость хранения единицы продукта;
- u_t — емкость (в единицах продукта) склада.

Нужно определить, сколько единиц продукта нужно поставлять в каждом из периодов, чтобы полностью удовлетворить спрос и суммарные затраты на поставку и хранение продукта были минимальны.

Процесс поиска оптимального плана поставок продукта естественно разбить на $m = T$ стадий. На стадии $t = 1, \ldots, T$ (здесь более естественно использовать индекс t вместо индекса i) мы решаем, сколько единиц продукта x_t нужно поставить в период t. Состояние s_t системы на стадии t — это количество продукта на складе в конце периода t. При таких обозначениях процесс принятия решений описывается следующим образом:

$$\begin{aligned}
S_t &\stackrel{\text{def}}{=} [0, u_t],\\
D_t(s_{t-1}) &\stackrel{\text{def}}{=} \mathbb{R}_+,\\
g_t(x_t, s_{t-1}) &\stackrel{\text{def}}{=} -c_t x_t - h_t(s_{t-1} + x_t - d_t) - f_t \operatorname{sign}(x_t),\\
tr_t(x_t, s_{t-1}) &\stackrel{\text{def}}{=} s_{t-1} + x_t - d_t.
\end{aligned}$$

Для рассматриваемой задачи рекуррентная формула (5.5) переписывается следующим образом:

$$\begin{aligned}
V_t(s_t) = \max\{&V_{t-1}(s_{t-1}) - c_t x_t - h_t(s_{t-1} + x_t - d_t) - f_t \operatorname{sign}(x_t) :\\
&s_t = s_{t-1} + x_t - d_t,\ x_t \geq 0,\ 0 \leq s_{t-1} \leq u_{t-1}\}, \quad t = 1, \ldots, T,
\end{aligned} \tag{5.14}$$

при начальных условиях $V_0(s_0) = 0$, $V_0(s) = -\infty$ для $s \neq s_0$. Вводя обозначения $H_t(s) \stackrel{\text{def}}{=} -V_t(s)$ для $t = 0, \ldots, T$, преобразуем формулу (5.14) к следующему виду:

$$\begin{aligned}
H_t(s_t) = \min\{&H_{t-1}(s_t + d_t - x_t) + c_t x_t + h_t s_t + f_t \operatorname{sign}(x_t) :\\
&\max\{0, s_t + d_t - u_{t-1}\} \leq x_t \leq s_t + d_t\}, \quad t = 1, \ldots, T,
\end{aligned} \tag{5.15}$$

при начальных условиях $H_0(s_0) = 0$, $H_0(s) = \infty$ для $s \neq s_0$. Заметим, что $H_t(s_t)$ есть стоимость оптимального плана в задаче о размере партии на плановом горизонте из t первых периодов, при начальном запасе продукта s_0 и остатке s_t после t периодов.

Замечание. Формулу (5.15) можно легко модифицировать для учета доголнительных ограничений. Скажем, если поставляется неделимый продукт, то в (5.15) нужно довавить ограничение $x_t \in \mathbb{Z}_+$. Чтобы быть готовым к неожиданному увеличению спроса на продукт, мы могли бы дополнительно потребовать, чтобы запас продукта в любой из периодов превосходил спрос в данный период как минимум на 10 %. Для этого, в формуле (5.15) достаточно наложить дополнительное ограничение на x_t: $s_{t-1} + x_t \geq 1.1 d_t$, или $x_t \geq 1.1 d_t - s_{t-1}$.

5.3.1. Неограниченная емкость склада

Когда емкость склада в любой из периодов превосходит суммарный остаточный (до конца планового горизонта) спрос, $u_t \geq d_{t,T}$, где $d_{t_1,t_2} \stackrel{\text{def}}{=} \sum_{\tau=t_1}^{t_2} d_t$, то можно считать, что емкость склада неограничена, и тогда $S_t = \mathbb{R}_+$. Предположим также, что $s_0 = 0$. При таких предположениях нетрудно доказать следующее утверждение:

Свойство 5.1. *Среди оптимальных решений задачи о размере партии существует такое решение (x^*, s^*), что $x_t^* s_{t-1}^* = 0$ для всех $t = 2, \ldots, T$. Пусть $x_{t_1}^*, x_{t_2}^*, \ldots, x_{t_k}^*$, где $t_1 < t_2 < \cdots < t_k$, есть все ненулевые компоненты вектора x^*. Тогда $x_{t_i}^* = d_{t_i, t_{i+1}}$ для $i = 1, \ldots, k$, где $t_{k+1} \stackrel{\text{def}}{=} T$.*

Из свойства (5.1) следует, что продукт производится только тогда, когда его запасы на складе полностью истощились. Из этого свойства также следует, что в конце планового горизонта запасы продукта на складе равны нулю.

Поскольку $s_t = \sum_{\tau=1}^{t} x_\tau - d_{1t}$, то

$$\begin{aligned} \sum_{t=1}^{T} (f_t y_t + c_t x_t + h_t s_t) &= \sum_{t=1}^{T} f_t y_t + \sum_{t=1}^{T} \left(c_t x_t + h_t \left(\sum_{\tau=1}^{t} x_\tau - d_{1t} \right) \right) \\ &= \sum_{t=1}^{T} (f_t y_t + w_t x_t) - \sum_{t=1}^{T} h_t d_{1t}, \end{aligned} \tag{5.16}$$

где $w_t \stackrel{\text{def}}{=} c_t + \sum_{\tau=t}^{T} h_\tau$. Равенство (5.16) позволяет свести задачу о размере партии с параметрами $\{f_t, c_t, h_t\}_{t=1}^{T}$ к задаче о размере партии с параметрами $\{\hat{f}_t = f_t, \hat{c}_t = w_t, \hat{h}_t = 0\}_{t=1}^T$, в которой все стоимости хранения продукта равны нулю.

Для $t = 1, \ldots, T$ пусть $\hat{H}_t$ обозначает стоимость оптимального решения для задачи о размере партии с параметрами $\{f_\tau, \hat{c}_\tau = w_\tau, \hat{h}_\tau = 0\}_{\tau=1}^t$. Заметим, что, определяя $\hat{H}_t$, мы не указываем запасы продукта на складе в конце периода t, поскольку при оптимальном планировании эти запасы равны нулю.

Используя свойсто 5.1, нетрудно убедиться в справедливости следующей рекуррентной формулы:

$$\begin{aligned} &\hat{H}_0 = 0, \\ &\hat{H}_t = \min_{1 \le \tau \le t} \{\hat{H}_{\tau-1} + f_\tau + w_\tau d_{\tau t}\}, \quad t = 1, \ldots, T. \end{aligned} \tag{5.17}$$

Пример 5.4. *Используя рекуррентную формулу (5.17), нужно решить задачу о размере партии со следующими параметрами:* $T = 4$, $d = (2, 4, 4, 2)^T$, $c = (3, 2, 2, 3)^T$, $h = (1, 2, 1, 1)^T$ *и* $f = (10, 20, 16, 10)^T$.

Решение. Начнем с вычисления значений $\hat{H}_t$, при этом, для каждого $t = 1, 2, 3, 4$ будем запоминать индекс τ_t, на котором достигается значение $\hat{H}_t$.

$$\begin{aligned}
\hat{H}_0 &= 0, \\
\hat{H}_1 &= \hat{H}_0 + f_1 + w_1 d_{11} = 0 + 10 + 8 \cdot 2 = 26, \quad \tau_1 = 1, \\
\hat{H}_2 &= \min\{\hat{H}_0 + f_1 + w_1 d_{12}, \hat{H}_1 + f_2 + w_2 d_{22}\} \\
&= \min\{0 + 10 + 8 \cdot 6, 26 + 20 + 6 \cdot 4\} = \min\{48, 70\} = 58, \quad \tau_2 = 1, \\
\hat{H}_3 &= \min\{\hat{H}_0 + f_1 + w_1 d_{13}, \hat{H}_1 + f_2 + w_2 d_{23}, \hat{H}_2 + f_3 + w_3 d_{33}\} \\
&= \min\{0 + 10 + 8 \cdot 10, 26 + 20 + 6 \cdot 8, 58 + 16 + 4 \cdot 4\} \\
&= \min\{90, 94, 90\} = 90, \quad \tau_3 = 1, \\
\hat{H}_4 &= \min\{\hat{H}_0 + f_1 + w_1 d_{14}, \hat{H}_1 + f_2 + w_2 d_{24}, \\
&\qquad \hat{H}_2 + f_3 + w_3 d_{34}, \hat{H}(3) + f_4 + w_4 d_{44}\} \\
&= \min\{0 + 10 + 8 \cdot 12, 26 + 20 + 6 \cdot 10, 58 + 16 + 4 \cdot 6, 90 + 10 + 4 \cdot 2\} \\
&= \min\{106, 106, 98, 108\} = 98, \quad \tau_4 = 3.
\end{aligned}$$

Теперь выполним обратный ход и найдем оптимальный план $x^* = (x_1^*, x_2^*, x_3^*, x_4^*)^T$. Поскольку значение $\hat{H}_4$ достигается при $\tau_4 = 3$, то $x_4^* = 0$, а $x_3^* = d_{34} = d_3 + d_4 = 6$. Далее, поскольку значение $\hat{H}_2$ достигается при $\tau = 1$, то $x_2^* = 0$, а $x_1^* = d_{12} = d_1 + d_2 = 6$.

Итак, оптимальный план следующий: $x^* = (6, 0, 6, 0)$. □

5.4. Контроль качества продукции, производимой на конвейере

На конвеере, на котором производится некоторый продукт, выполняется n операций. После каждой операции возможен контроль качества. Продукт поступает на конвеер партиями по $B \geq 1$ штук в одной партии. После выполнения каждой операции могут появиться дефекты, которые нельзя исправить. Поэтому дефектные экземпляры выбрасываются в отходы. Пусть p_i — вероятность появления дефекта при выполнении операции i. После выполнения ряда операций (каких конкретно нужно будет определить) проводится контроль качества всей партии (выборочный контроль не проводится). Будем считать, что контроль определяет все дефектные экземпляры. Нужно найти оптимальный план контроля качества, который указывает после каких операций проводить контроль. Меньшее количество инспекций уменьшает затраты на их проведение, но увеличивает производственные издержки из-за того, что могут выполняться ненужные операции на дефектных экземплярах.

Заданы следующии параметры:

- α_i — стоимость выполнения операции i на одном экземпляре;
- f_{ij} — фиксированная стоимость контроля одной партии продукта после операции j, при условии, что предшествующий контроль проводился после операции i;
- g_{ij} — стоимость контроля единицы продукта после операции j, при условии, что предшествующий контроль проводился после операции i.

Стоимость контроля после операции j зависит от того, когда проводился предшествующий контроль. Если это было после операции i, то контроль должен определить все дефекты, которые были сделаны на всех промежуточных этапах $i+1, i+2, \ldots, j$. Поэтому стоимость контроля после операции j, при условии что предшествующий контроль проводился после операции i, равна

$$c(i,j) \stackrel{\text{def}}{=} f_{i,j} + B(i)g_{ij} + B(i)\sum_{k=i+1}^{j} \alpha_k\,, \tag{5.18}$$

где $B(i) = B\prod_{k=1}^{i}(1-p_k)$ есть ожидаемое количество недефектных единиц продукта после операции i. Первые два слагаемых в (5.18) задают стоимость контроля после операции j, при условии что предшествующий контроль проводился после операции i, а третий член суммы —

это суммарные производственные издержки при выполнении операций $i{+}1, i{+}2, \ldots, j$. Определим $F(j)$ минимальные суммарные издержки контроля качества, если для производства изделия нужно последовательно выполнить операции $1, 2, \ldots, j$. Величины $F(j)$ можно вычислить по следующей рекурентной формуле:

$$\begin{aligned} &F(0) = 0, \\ &F(j) = \min_{0 \le i < j} (F(i) + c(i,j)), \quad j = 1, \ldots, n. \end{aligned} \tag{5.19}$$

5.5. Модель оптимального роста Касса — Купманса

Агрегированное предпочтение всех домашних хозяйств представляется *функцией полезности* $U : \mathbb{R} \to \mathbb{R}$. Рассмотрим бесконечный временной интервал и предположим, что полезности не зависят от предистории, т. е. полезность $U(c_t)$ в период $t = 0, 1, 2, \ldots$ зависит только от уровня потребления c_t в период t,

Один и тот же уровень потребления в двух последовательных периодах домашние хозяйства оценивают по разному, справедливо считая, что с течением времени потребление должно расти. Поэтому полезность $U(c_{t+1})$ в периоде $t+1$ с точки зрения периода t оценивается как $\beta U(c_{t+1})$, где β $(0 < \beta < 1)$ есть *дисконтный множитель*.

Предположим, что технология в период 0 такова, что, используя капитал k, можно произвести продукции на сумму $y = f(k)$. Напомним, что в микроэкономике функция f, связывающая затраты и выпуск, называется *производственной функцией*. В данной простой модели предполагается, что с течением времени технология не меняется (производственная функция одна и та же во всех периодах). Мы будем обозначать через k_t капитал, доступный домашним хозяйствам в начале периода t. Соответственно, $y_t = f(k_t)$ есть стоимость выпуска в период t. Часть капитала y_t потребляется в периоде t, а оставшаяся часть инвестируется в производство в следующем периоде $t+1$:

$$f(k_t) = c_t + k_{t+1}.$$

С учетом всех сделанных выше допущений, *задача социального пла-*

нирования формулируется следующим образом:

$$\sum_{t=0}^{\infty} \beta^t U(c_t) \to \max, \tag{5.20a}$$

$$f(k_t) = c_t + k_{t+1}, \quad t = 0, 1, 2, \ldots, \tag{5.20b}$$

где величина начального капитала k_0 фиксирована.

Содержательно, задача (5.20) состоит в том, чтобы определить уровни потребления и инвестировования на бесконечном временном горизонте так, чтобы суммарная по всем периодам дисконтированная полезность домашних хозяйств была максимальной.

5.5.1. Рекуррентная формула

До сих пор мы рассматривали применения динамического программирования для решения комбинаторных задач, характерной чертой которых является то, что их можно решить перебором конечного (иногда очень большого) количества вариантов решения. Здесь мы имеем несколько иную ситуацию, в которой мы даже не можем записать решение задачи, перечислив все решения, которые нужно принимать в каждом периоде, которых бесконечное количество. Единственное, на что мы можем надеяться, — это найти формулу (в общем случае рекуррентную) для вычисления уровня капитала k_t в каждом из временных периодов $t = 1, 2, \ldots$

Определим

$$\begin{aligned} v(q) \stackrel{\text{def}}{=} \max_{\{k_t\}_{t=1}^{\infty}} &\sum_{t=0}^{\infty} \beta^t U(c_t), \\ &f(k_t) = c_t + k_{t+1}, \quad t = 0, 1, 2, \ldots, \\ &k_0 = q. \end{aligned}$$

Тогда

$$\begin{aligned}
v(k_0) &= \max_{\{k_t\}_{t=1}^{\infty}} \left(\sum_{t=0}^{\infty} \beta^t U(c_t) : f(k_t) = c_t + k_{t+1},\ t = 0, 1, 2, \ldots \right) \\
&= \max_{\{k_t\}_{t=1}^{\infty}} \left(U(c_0) + \sum_{t=1}^{\infty} \beta^t U(c_t) : f(k_t) = c_t + k_{t+1},\ t = 0, 1, 2, \ldots \right) \\
&= \max_{k_1} \left(U(c_0) + \beta v(k_1) : f(k_0) = c_0 + k_1 \right) \\
&= \max_{k_1} \left(U(f(k_0) - k_1) + \beta v(k_1) \right). \qquad (5.21)
\end{aligned}$$

Условие оптимальности первого порядка для оптимизационной задачи (5.21), записывается следующим образом:

$$U'(f(k_0) - k_1) = \beta v'(k_1). \qquad (5.22)$$

Это равенство означает, что предельная полезность в начальный период 0 при уровне потребления $c_0 = f(k_0 - k_1)$ должна быть равна дисконтированной предельной суммарной полезности при начальном капитале k_1.

Далее мы попытаемся преобразовать условие оптимальноси (5.22), чтобы получить более полезное условие оптимальности, содержащее только известные функции U и f, но не содержащее неизвестную функцию v. Предположим, что задача (5.21) имеет решение $k_1 = g(k_0)$. Тогда

$$v(k_0) = U(f(k_0) - g(k_0)) + \beta v(g(k_0)).$$

Дифференцируя это равенство по k_0, получим

$$v'(k_0) = U'(f(k_0) - g(k_0))(f'(k_0) - g'(k_0)) + \beta v'(g(k_0))g'(k_0).$$

Используя (5.22), получим

$$\begin{aligned}
v'(k_0) &= U'(f(k_0) - g(k_0))f'(k_0) - \left(U'(f(k_0) - g(k_0)) - \beta v'(g(k_0))\right) g'(k_0)) \\
&= U'(f(k_0) - k_1)f'(k_0).
\end{aligned}$$

Подставляя k_t вместо k_0, получим равенства

$$v'(k_t) = U'(f(k_t) - k_{t+1})f'(k_t), \quad t = 0, 1, 2, \ldots$$

Теперь подставим $v'(k_1)$ в (5.22):

$$U'(f(k_0) - k_1) = \beta U'(f(k_1) - k_2)f'(k_1).$$

Снова, подставляя k_t вместо k_0, получим равенства

$$U'(f(k_t) - k_{t+1}) = \beta U'(f(k_{t+1}) - k_{t+2})f'(k_{t+1}), \quad t = 1, 2, \ldots \qquad (5.23)$$

5.5.2. Специальный случай функции полезности

Рассмотрим частный случай задачи (5.20) с производственной функцией *Коба — Дугласа* $f(k) = k^\alpha$ и *логарифмической* функцией полезности $U(c) = \ln c$. В данном случае условие оптимальности (5.23) переписывается в следующем виде:

$$\frac{1}{k_t^\alpha - k_{t+1}} = \alpha\beta \frac{k_{t+1}^{\alpha-1}}{k_{t+1}^\alpha - k_{t+2}}.$$

С учетом равенства $f(k_t) = k_t^\alpha = c_t + k_{t+1}$ имеем

$$\frac{k_{t+1}}{c_t} = \alpha\beta \frac{k_{t+1}^\alpha}{c_{t+1}}$$

и

$$\frac{k_t^\alpha}{c_t} = 1 + \frac{k_{t+1}}{c_t} = 1 + \alpha\beta \frac{k_{t+1}^\alpha}{c_{t+1}}.$$

Далее, продвигаясь на один период вперед, получим равенство:

$$\frac{k_t^\alpha}{c_t} = 1 + \alpha\beta \frac{k_{t+1}^\alpha}{c_{t+1}} = 1 + \alpha\beta \left(1 + \alpha\beta \frac{k_{t+2}^\alpha}{c_{t+2}}\right).$$

Продолжая так двигаться до бесконечности, получим

$$\frac{k_t^\alpha}{c_t} = 1 + \alpha\beta + (\alpha\beta)^2 + (\alpha\beta)^3 + \cdots = \frac{1}{1 - \alpha\beta}.$$

Откуда

$$c_t = (1 - \alpha\beta) k_t^\alpha \quad \text{и} \quad k_{t+1} = \alpha\beta k_t^\alpha.$$

5.6. Упражнения

5.1. Нужно представить целое положительное число b в виде суммы n целых положительных чисел $b = \sum_{i=1}^n x_i$, так, чтобы их произведение $\prod_{i=1}^n x_i$ было максимальным. Пусть $f_n(b)$ обозначает максимальное значение произвения. Запишите рекуррентное соотношение для $f_n(b)$ и по нему найдите значение $f_n(b)$.

5.2. *Агрегация систем линейных уравнений с целыми коэффициентами.* Рассмотрим систему уравнений

$$\sum_{j=1}^n a_{ij} x_j = b_i, \quad i = 1, 2, \tag{5.24}$$

с целыми неотрицательными коэффициентами a_{ij}, b_1 и b_2. Докажите следующую теорему.

Теорема 5.1. *Пусть λ_1 и λ_2 есть взаимно простые целые числа, причем λ_1 не делит b_2, а λ_2 не делит b_1. Если $\lambda_1 > b_2 - a^{\min}$ и $\lambda_2 > b_1 - a^{\min}$, где $a^{\min}$ есть наименьшее из ненулевых чисел a_{ij}, то множество неотрицательных целочисленных решений системы* (5.24) *совпадает с множеством неотрицательных целочисленных решений их линейной комбинации*

$$\sum_{j=1}^{n}(\lambda_1 a_{1j} + \lambda_2 a_{2j})x_j = \lambda_1 b_1 + \lambda_2 b_2.$$

5.3. Решите следующие задачи о рюкзаке:

а) $$\begin{aligned} 15x_1 + 19x_2 + 24x_3 + 27x_4 &\to \max \\ 2x_1 + 3x_2 + 4x_3 + 5x_4 &\le 7, \\ x_1, x_2, x_3, x_4 &\in \mathbb{Z}_+; \end{aligned}$$

б) $$\begin{aligned} 12x_1 + 5x_2 + 17x_3 + 9x_4 + 7x_5 &\to \max \\ 3x_1 + 5x_2 + 2x_3 + 4x_4 + 6x_5 &\le 9, \\ x_1, x_2, x_3, x_4, x_5 &\in \{0,1\}; \end{aligned}$$

в) $$\begin{aligned} x_1 + 2x_2 + 3x_3 + 2x_4 + 3x_5 &\to \max \\ 12x_1 + 9x_2 + 15x_3 + 11x_4 + 6x_5 &\le 29, \\ x_1, x_2, x_3, x_4, x_5 &\in \{0,1\}. \end{aligned}$$

5.4. Корпорация выделяет 10 миллионов долларов для расширения производства на трех своих предприятиях. Предприятия представили на рассмотрение 5 проектов, информация по которым приведена в следующей таблице:

Проект	Предприятие 1		Предприятие 2		Предприятие 3	
	Стои-мость	ожид. доход	Стои-мость	ожид. доход	Стои-мость	ожид. доход
1	1	5	2	8	1	4
2	2	6	3	10	0	0
3	0	0	4	12	0	0
4	3	8	0	0	2	5
5	0	0	2	5	1	3

Например, проект 2 совместно реализуют предприятия 1 и 2, причем предприятию 1 нужно выделить \$2 млн., а предприятию 2 — \$3 млн.; после реализации проекта в течении последующих пяти лет на предприятии 1 ожидается доход \$6 млн., а на предприятии 2 — \$10 млн.

Какие проекты нужно реализовать, чтобы суммарный ожидаемый доход был максимален?

5.5. *Многомерная задача о рюкзаке* — это задача ЦП

$$\max\{c^T x : \ Ax \le b, \ x \in \mathbb{Z}_+^n\}, \tag{5.25}$$

в которой все числовые параметры неотрицательные и целые. Запишите рекуррентную формулу для решения задачи (5.25) методом динамического программирования. Оцените временную и емкостную (объем памяти) сложность вычислений по вашей рекуррентной формуле.

5.6. Решить примеры задачи о размере партии со следующими параметрами:

а) $T = 5$, $d = (2,4,4,2,3)^T$, $c = (3,2,2,3,1)^T$, $h = (1,2,1,1,2)^T$ и $f = (10,20,16,10,8)^T$,

б) $T = 5$, $d = (4,2,3,4,3)^T$, $c = (2,3,2,3,2)^T$, $h = (1,2,2,1,2)^T$ и $f = (10,12,20,10,10)^T$.

5.7. Требуется определить, когда в течении временного горизонта из T лет нужно заменять автомобиль на новый, чтобы минизировать общие затраты на покупку и эксплуатацию автомобиля. Предполагается, что решение о покупке автомобиля принимается в начале каждого года, и автомобиль не может эксплуатироваться более n $(n < T)$ лет. В начале планового горизонта 1) владелец имеет автомобиль возраста τ_0, 2) стоимость нового автомобиля равна p, 3) для автомобиля возраста k $(k = 1, \dots, n)$, расходы по эксплуатации в течении года равны c_k $(c_1 < c_2 < \dots < c_n)$, а его остаточная стоимость равна r_k $(r_1 > r_2 > \dots > r_n,\ r_1 > 0,\ r_n < 0)$. Предполагается, что в течении планового горизонта уровень годовой инфляции постоянен и равен q процентов.

Обозначим через $E(\tau, t)$ минимальные издержки эксплуатации автомобиля начального возраста τ в течении t лет, $\tau = 0, \dots, n$, $t = 0, \dots, T$. Начальные условия: $E(\tau, 0) = 0$ для всех $\tau = 0, \dots, n$. Запишите рекуррентную формулу для вычисления значений $E(\tau, t)$.

Глава 6
Методы анализа сетей

Сетью называется орграф, вершинам и дугам которого приписаны числовые параметры. Понятно, что в значительной степени сетевой анализ основан на теории графов, необходимые сведения из которой приведены в приложении C. Сетевые модели широко используются в исследовании операций. На практике сети являются естественными моделями таких объектов как транспортные, электрические и компьютерные сети, системы телекоммуникаций и водоснабжения, трубопроводы и др.

6.1. Кратчайшие пути

Дан ориентированный граф $G = (V, E)$, каждой дуге которого приписаны *стоимость* (*длина*) $c(v, w)$. Существуют несколько классических формулировок задач поиска кратчайших путей в графах.

1. **Кратчайший путь между двумя выделенными вершинами**: в графе G нужно найти путь

$$P = \{s = v_0, v_1, \ldots, v_{k-1}, v_k = t\}$$

от вершины $s \in V$ до вершины $t \in V$ минимальной стоимости (*кратчайший путь*)

$$c(P) \stackrel{\text{def}}{=} \sum_{i=1}^{k} c(v_{i-1}, v_i)\,.$$

2. **Кратчайший пути от одной выделенной вершины до всех остальных**: в графе G нужно найти кратчайшие пути от выделенной вершины $s \in V$ до всех остальных вершин графа.

3. **Кратчайшие пути между всеми парами вершин**: в графе G нужно найти кратчайшие пути между каждой парой вершин $s, t \in V$.

Методы поиска кратчайших путей в графах используются также в более сложных задачах сетевой оптимизации. Можно также утверждать, по меньшей мере в отношении комбинаторных задач[10], что методы динамического программирования являются не чем иным как алгоритмами поиска кратчайших путей на некоторых графах.

6.1.1. Дерево кратчайших путей

Как это не парадоксально, но найти кратчайший путь между двумя выделенными вершинами не легче, чем искать кратчайшие пути от одной вершины до всех остальных.

Пусть $G = (V, E)$ есть орграф с выделенной вершиной $s \in V$, Каждой дуге $(v, w) \in E$ приписана стоимость $c(v, w)$. Без ограничения общности будем считать, что из s *достигаются* (существует путь во) все остальные вершины графа. Если это не так, то мы можем добавить к G дуги (s, v), $v \in V \setminus s$, бесконечной стоимости $c(s, v) = \infty$. Стоимость кратчайшего пути в графе G от вершины s до вершины $v \in V$ обозначим через $\sigma(s, v)$ (если такого пути не существует, то $\sigma(s, v) = +\infty$).

Пусть $P = (s = v_0, v_1 \ldots, v_k = v)$ есть кратчайший путь от s до v. Тогда для любого $0 \le i \le k$ стоимость подпути $P_i = (s = v_0, v_1 \ldots, v_i)$ пути P равна $\sigma(s, v_i)$, так как иначе отрезок P_i пути P можно заменить кратчайшим путем из s в v и получить более короткий путь P' из s в v. Это есть *принцып оптимальности*. Если в графе G нет циклов отрицательной стоимости, то все кратчайшие пути являются простыми и из принцыпа оптимальности можно сделать вывод, что стоимости кратчайших путей удовлетворяют следующим уравнениям Беллмана:

$$\begin{aligned} &\sigma(s, s) = 0, \\ &\sigma(s, v) = \min_{(w,v)\in E} (\sigma(s, w) + c(w, v)) \quad \text{для всех } v \in V \setminus s. \end{aligned} \tag{6.1}$$

Следующие два понятия, которые идут от линейного программирования, играют фундаментальную роль во всей потоковой оптимизации. *Функция цен* (*вектор потенциалов*) есть функция $p : V \to \mathbb{R}$. *Приведенная функция стоимости относительно функции цен* p определяется по правилу:

$$c_p(v, w) = c(v, w) + p(v) - p(w).$$

[10] *Комбинаторными* называют задачи, которые можно решить перебором конечного (возможно очень большого) количества вариантов решения.

Цены вершин имеют натуральную экономическую интерпретацию как действующие рыночные цены на некоторый продукт. Мы можем интерпретировать приведенную стоимость $c_p(v, w)$, как суму затрат на закупку единицы продукта в вершине v по цене $p(v)$ и затрат на транспортировку в вершину w минус доход от продажи ее там по цене $p(w)$.

Лемма 6.1. *Пусть $G = (V, E)$ есть орграф, на дугах которого определена функция стоимости c, а на вершыннах функция цен p. Тогда для пути P из вершины v в вершыну w в графе G имеет место равенство $c_p(P) = c(P) + p(v) - p(w)$. В частности, если P — цикл, то $c_p(P) = c(P)$.*

Доказательство. Пусть $P = (v = v_0, v_1, \ldots, v_{k-1}, v_k = w)$. Тогда

$$
\begin{aligned}
c_p(P) &= \sum_{i=1}^{k} c_p(v_{i-1}, v_i) = \sum_{i=1}^{k} (c(v_{i-1}, v_i) + p(v_{i-1}) - p(v_i)) \\
&= \sum_{i=1}^{k} c(v_{i-1}, v_i) + \sum_{i=1}^{k} p(v_{i-1}) - \sum_{i=1}^{k} p(v_i) \\
&= c(P) + p(v) - p(w).
\end{aligned}
$$

□

Для покрывающего ордерева T с корнем s *функция расстояний* $d : V \to \mathbb{R}$ определяется рекурсивно следующим образом:

$$d(s) = 0,\ d(v) = d(parent(v)) + c(parent(v), v) \quad \text{для } v \in V \setminus s\,,$$

где $parent(v)$ есть *отец* (начальная вершина единственной дуги, которая входит в v) вершины v в дереве T. Покрывающее ордерево T с корнем s называется *деревом кратчайших путей*, если для каждой вершины v единственный путь в дереве T из s в v является кратчайшим путем из s в v в графе G, т. е. $d(v) = \sigma(s, v)$.

Теорема 6.1. *Пусть все вершины графа $G = (V, E)$ достигаются из вершины $s \in V$. Граф G имеет дерево кратчайшийх путей тогда и только тогда, когда он не имеет циклов отрицательной стоимости. Покрывающее ордерево T с корнем s является деревом кратчайших путей тогда и только тогда, когда его функция расстояний d удовлетворяет условию:*

$$c_d(v, w) \geq 0 \quad \text{для всех } (v, w) \in E. \tag{6.2}$$

Доказательство. Если G не имеет циклов отрицательной стоимости, то кратчайшие пути от s до всех остальных вершин являются простыми. Пусть D есть объединение дуг этих путей. Очевидно, что граф (V, D) содержит покрывающее ордерево T, которое является деревом кратчайших путей.

Понятно, что функция расстояний дерева кратчайших путей удовлетворяет условию (6.2). Докажем обратное. Пусть дерево T удовлетворяет условию (6.2). По лемме 6.1 для любого цикла Γ графа G имеем $c(\Gamma) = c_d(\Gamma) \geq 0$. Пусть $s = v_0, v_1, \ldots, v_k = v$ есть кратчайший путь из s в v. Так как G не имеет циклов отрицательной стоимости, то стоимость кратчайшего пути из s в s равна $0 = d(s)$. По индукции допустим, что $d(v_{k-1})$ есть стоимость кратчайшего пути из s d v_{k-1}. Так как по (6.2) $d(v) \leq d(v_{k-1}) + c(v_{k-1}, v)$, а $d(v_{k-1}) + c(v_{k-1}, v)$ есть стоимость кратчайшего пути из s в v, то $d(v)$ — также стоимость кратчайшего пути из s в v. □

Следствие 6.1. *Орграф $G(V, E)$ не имеет циклов отрицательной стоимости тогда и только тогда, когда существует функция цен $p : V \to \mathbb{R}$, что $c_p(v, w) \geq 0$ для всех $(v, w) \in E$.*

Доказательство. Достаточность условия (6.2) утверждается в лемме 6.1. Докажем необходимость. Построим вспомогательный орграф

$$G_{\mathrm{aux}} = (V_{\mathrm{aux}}, E_{\mathrm{aux}}) = (V \cup \{s\}, E \cup (\{s\} \times V)),$$

добавляя к G новую вершину s и множество дуг, которые выходят из s во все остальные вершины. Стоимости новых дуг определим равными нулю. Очевидно, если G не имеет циклов отрицательной стоимости, то их нет и в G_{aux}. Поэтому по теореме 6.1 граф G_{aux} имеет дерево кратчайших путей и его функция расстояний d удовлетворяет условию (6.2). Нам осталось только положить $p = d$. □

6.1.2. Алгоритм построения дерева кратчайших путей

Идея всех алгоритмов поиска кратчайших путей в графах одинакова. Все они начинают с функций $d : V \to \mathbb{R}$ и *parent* $: V \to V \cup \{\mathbf{nil}\}$, таких, что

$$d(v) = \begin{cases} 0, & v = s, \\ \infty, & v \in V \setminus \{s\}, \end{cases}$$
$$parent(v) = \mathbf{nil}, \quad v \in V.$$

Затем итеративно повторяется следующий шаг:

> выбрать дугу (v, w) отрицательной приведенной стоимости $c_d(v, w) < 0$, положить $parent(w) = v$ и заменить $d(w)$ на $d(v) + c(v, w)$.

Это есть *метод последовательной аппроксимации.*

Если в графе нет циклов отрицательной стоимости, то в случае, когда стоимости дуг целочисленны, после конечного числа итераций метод заканчивает работу. При этом $d(v) = \sigma(s, v)$ для всех $v \in V$, а указатели $parent$ задают дерево кратчайших путей.

Если в графе есть цикл отрицательной стоимости, достижимый из вершины s, то метод должен остановиться, как только граф, составленный из дуг $(parent(v), v)$ с $parent(v) \neq \mathbf{nil}$, содержит цикл. Из описания метода последовательной аппроксимации следует, что стоимость этого цикла отрицательна.

6.1.3. Алгоритм Форда — Беллмана

Эффективность метода последовательной аппроксимации существенно зависит от порядка выбора дуг отрицательной приведенной стоимости. В этом параграфе мы будем рассматривать алгоритм, который предложен независимо Фордом и Беллманом. Процедура $FordBellman$, которая реализует этот алгоритм, представлена в листинге 6.1. Если процедура возвращает значение **истина**, то указатели $parent$ задают дерево кратчайшийх путей, а функция d есть его функция расстояний. Если же процедура возвращает значение **ложь**, то любой цикл подграфа $(V, \overline{E})$, где

$$\overline{E} \stackrel{\text{def}}{=} \{(parent(v), v) \,:\, v \in V,\ parent(v) \neq \mathbf{nil}\},$$

является циклом отрицательной стоимости в графе G. Чтобы гарантироваць, что все вершины графа G достигаются из s, если $(s, v) \notin E$, мы добавляем (условно) эту дугу к G и приписываем ей стоимость ∞. Если после завершения алгоритма $d(v) = \infty$, то в графе G вершина v недостигается из s.

Обозначим через $\sigma^i(s,v)$ стоимость кратчайшего пути из s в v среди всех путей, которые имеют ровно i дуг. Если такого пути не существует, то $\sigma^i(s,v)=\infty$. Пусть $d^i(v)$, S^i — соответственно $d(v)$ и список S после итерации i. Этап инициализации (шаги 1 и 2) называем 0-й итерацией.

Вход: Орграф $G=(V,E)$, функция $c: E\to\mathbb{R}$, вершина $s\in V$.
Выход: указатели $parent: V\to V\cup\{\mathbf{nil}\}$, функция $d: V\to\mathbb{R}$:
а) если указатели $parent$ представляют дерево, то
$d(v)$ — *кратчайшее расстояние* от s до v;
б) в противном случае, любой цикл в графе, представленном
$parent$, является *отрицательным циклом.*
1. Для всех $v\in V\setminus\{s\}$ положить $d(v)=\infty$ и $parent(v)=\mathbf{nil}$.
2. Положить $d(s)=0$ и $S=\{s\}$.
3. Для $i=1,\dots,n$ выполнить шаги 3.1–3.4:
 3.1. $Q=\emptyset$, $dw=d$.
 3.2. Для $(v,w)\in E(S,V)$, таких, что $d(w)>dw(v)+c(v,w)$,
 положить $d(w):=dw(v)+c(v,w)$, $parent(w)=v$, $Q\leftarrow w$.
 3.3. Если $Q=\emptyset$, вернуть **истина**.
 3.4. Положить $S=Q$.
4. Вернуть **ложь**.

Листинг 6.1. Алгоритм Форда — Беллмана

Лемма 6.2. *Справедливы следующие соотношения:*

а) $d^i(v)=\min_{0\le k\le i}\sigma^k(s,v)$ *для всех* $v\in V$*,*

б) $d^i(v)=\sigma^i(s,v)<\sigma^{i-1}(s,v)$ *для всех* $v\in S^i$*,*

Доказательство. Очевидно, что утверждение леммы справедливо после 0-й итерации. Допустим, что после завершения $(i-1)$-й (перед началом i-й) итерации алгоритма

$$d^{i-1}(v)=\min_{0\le k\le i-1}\sigma^k(s,v)\quad \text{для всех } v\in V\,,$$
$$d^{i-1}(v)=\sigma^{i-1}(s,v)\quad \text{для всех } v\in S^{i-1}.$$

Пусть $w\in V$ и $s=v_0,v_1,\dots,v_{i-1},v_i=w$ есть путь стоимости $\sigma^i(s,w)$ у графе G. Тогда по допущению $d^{i-1}(v_{i-1})=\sigma^{i-1}(v_{i-1})$. Поэтому $v_{i-1}\in S^{i-1}$ и, если $d^{i-1}(w)>d^{i-1}(v_{i-1})+c(v_{i-1},w)=\sigma^i(s,w)$, то после i-й итерации $d^i(w)=\sigma^i(s,w)$, а вершина w будет включена в S_i. □

Лемма 6.3. *Если после завершения алгоритма Форда — Беллмана $S = \emptyset$, то функция расстояний d удовлетворяет условию (6.2).*

Доказательство. Заметим, что $d^i(v) \geq d^{i+1}(v)$ для всех $v \in V$. Пусть $(v, w) \in E$. Так как $S = \emptyset$, то $d(v) = d^i(v)$ для некоторого $1 \leq i \leq n-1$. Поэтому

$$d(w) \leq d^{i+1}(w) \leq d^i(v) + c(v, w) = d(v) + c(v, w).$$

□

Лемма 6.4. *Граф G имеет цикл отрицательной стоимости тогда и только тогда, когда после завершения алгоритма Форда — Беллмана множество S не пустое.*

Доказательство. Если $S = \emptyset$, то по теореме 6.1 и лемме 6.3 граф G не имеет циклов отрицательной стоимости. Если $S \neq \emptyset$, то по лемме 6.2 для $v \in S$

$$d^n(v) = \sigma^n(s, v) < \sigma^i(s, v) \quad \text{для всех } 0 \leq i < n.$$

Очевидно, что каждый путь из s в v длины n и стоимости $\sigma^n(s, v)$ обходит цикл отрицательной стоимости. □

Теорема 6.2. *За время $O(nm)$ алгоритм Форда — Беллмана или строит дерево кратчайшийх путей, или находит цикл отрицательной стоимости.*

Доказательство. Корректность алгоритма вытекает из лемм 6.3 и 6.4 Так как сложность одной итерации процедуры алгоритма Форда — Беллмана не превосходит $O(m)$, а количество итераций не больше n, то общая сложность процедуры алгоритма есть $O(nm)$. □

Пример 6.1. *В графе, представленном на рис. 6.1, найти кратчайшие пути от вершины* 1 *до всех остальных вершин.*

Работа алгоритма по итерациям представлена в табл. 6.1. Этап инициализации есть итерация 0. Функция dw — это функция d на предыдущейитерации, а множество Q — это множество S на следующей итерации. Дерево кратчайших путей задается функцией *parent* на последней 5-й итерации. Его дуги представлены на рис. 6.1 жирными линиями. □

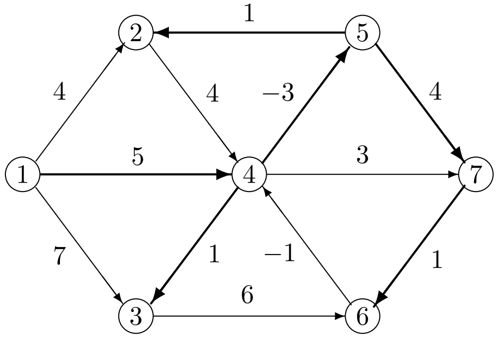


Рис. 6.1. Граф для примера 6.1

Таблица 6.1

Итерации алгоритма Форда — Беллмана

Ит.	S	d							$parent$						
		1	2	3	4	5	6	7	1	2	3	4	5	6	7
0	1	0	∞	∞	∞	∞	∞	∞	-	-	-	-	-	-	-
1	$2, 3, 4$	0	4	7	5	∞	∞	∞	-	1	1	1	-	-	-
2	$3, 5, 6, 7$	0	4	6	5	2	13	8	-	1	4	1	4	3	4
3	$2, 6, 7$	0	3	6	5	2	9	6	-	5	4	1	4	7	5
4	6	0	3	6	5	2	7	6	-	5	4	1	4	7	5
5	$\emptyset$	0	3	6	5	2	7	6	-	5	4	1	4	7	5

Замечание. При описании алгоритма Форда — Беллмана мы ввели функцию dw только для того, чтобы получить соотношения а) и б) из леммы 6.2. Понятно, что сложность алгоритма не увеличиться, если вместо dw использовать функцию d. Наоборот, в среднем алгоритм будет работать быстрее.

Циклы отрицательной стоимости

Задача поиска в орграфе $G = (V, E)$, дугам $(v, w) \in E$ которого приписаны стоимости $c(v, w)$, цикла Γ отрицательной стоимости $c(\Gamma) < 0$, имеет самостоятельный как практический так теоретический интерес.

Мы знаем, что алгоритм Форда — Беллмана может найти отрицательный цикл в графе, если этот цикл достижим из стартовой вершины. Чтобы шарантировать это, добавим к графу G новую вершину s и соединим ее дугой (s, v) нулевой стоимости с каждой вершиной $v \in V$. В расширенном графе G' 1) каждая вершина достижима из вершины s, и 2) любой цикл в G' также является циклом в G (поскольку в s не входят дуги, то вершина s не принадлежит ни одному циклу). Применим алгоритм Форда — Беллмана к графу G' с начальной вершиной s, и, если в G есть отрицательный цикл, то алгоритм найдет его.

Арбитраж на валютном рынке

Все валюты, представленные на некотором валютном рынке, соответствуют вершинам $v \in V$ графа $G = (V, E)$. Каждой дуге $(v, w) \in E$, представляющей транзакцию валюты v в валюту w, приписан обменный курс $\gamma(v, w)$: единицу валюты v можно обменять на $\gamma(v, w)$ единиц валюты w. Цикл $\Gamma = (v_0, v_1, \ldots, v_k = v_0)$ в графе G называется *арбитражем на валютном рынке*, если произведение обменных курсов на дугах этого цикла больше единицы: $\prod_{i=1}^{k} \gamma(v_{i-1}, v_i) > 1$.

Задача поиска арбитража на валютном рынке сводится к задаче поиска цикла отрицательной стоимости в графе G, если его дугам приписать стоимости $c(v, w) = -\ln(\gamma(v, w))$. Действительно, стоимость цикла $\Gamma = (v_0, v_1, \ldots, v_k = v_0)$ в графе G равна

$$c(\Gamma) = \sum_{i=1}^{k} c(v_{i-1}, v_i) = -\sum_{i=1}^{k} \ln(\gamma(v_{i-1}, v_i)) = -\ln\left(\prod_{i=1}^{k} \gamma(v_{i-1}, v_i)\right).$$

Поэтому из $c(\Gamma) < 0$ следует, что $\prod_{i=1}^{k} \gamma(v_{i-1}, v_i) > 1$.

6.1.4. Алгоритм Дейкстры

Существует ряд ситуаций, в которых задача поиска кратчайших путей решается особенно просто. Одна из таких ситуаций, когда стоимости всех дуг неотрицательные. В этом случае для построения дерева кратчайших путей существует более эффективный (чем алгоритм Форда — Беллмана) алгоритм Дейкстры, который приведен в листинге 6.2. На каждой стадии алгоритм делит множество вершин V на два подмножества: S и $V \setminus S$. Если вершина v принадлежит S, то кратчайший путь до ее уже найден и его длина равна $d(v)$. На очередной итерации алгоритм выбирает вершину $w \in V \setminus S$ с минимальной «меткой» $d(w)$, добавляе ее к S, и для всех дуг $(w, v) \in E(w, V \setminus S$ перевычисляет метки их конечных вершин по правилу:

$$d(v) = \min\{d(v), d(w) + c(w, v)\}.$$

Вход: Орграф $G = (V, E)$, функция $c : E \to \mathbb{R}$, вершина $s \in V$.
Выход: указатели $parent : V \to V \cup \{\mathbf{nil}\}$, представляющие дерево кратчайших путей, функция $d : V \to \mathbb{R}$, где $d(v)$ — *кратчайшее расстояние* от s до v.

1. Для всех $v \in V \setminus \{s\}$ положить $d(v) = \infty$ и $parent(v) = \mathbf{nil}$.
2. Положить $d(s) = 0$ и $S = \emptyset$.
3. Пока $|S| < n - 1$ выполнять шаги 3.1 и 3.2:
   3.1. Выбрать $w \in \arg\min\{d(v) : v \notin S\}$ и положить $S := S \cup \{w\}$.
   3.2. Для всех $(w, v) \in E(w, S)$, таких, что $d(v) > d(w) + c(w, v)$, положить $d(v) = d(w) + c(w, v)$ и $parent(v) = w$.

Листинг 6.2. Алгоритм Дейкстры

Лемма 6.5. *Алгоритм Дейкстры поддерживает следующие инварианты:*

а) $d(v) \le d(w)$ *для всех* $v \in S$, $w \in V \setminus S$;

б) $d(v) + c(v, w) \ge d(w)$ *для всех* $(v, w) \in E(V, S)$.

Доказательство. Индукцией по $|S|$. Детали оставляем читателю. □

Теорема 6.3. *Если стоимости всех дуг неотрицательны, то за время* $O(n^2)$ *алгоритм Дейкстры строит дерево кратчайшийх путей.*

Доказательство. Корректность алгоритма следует из леммы 6.5. Оценим сложность алгоритма. На этапе инициализации требуется $O(n)$ операций. Сложность одной итерации $O(n)$. А так как количество всех итераций равно $n - 1$, то сложность всего алгоритма — $O(n^2)$. □

Пример 6.2. *В графе, представленном на рис. 6.2, нужно найти кратчайшие пути от вершины* 1 *до всех остальных вершин.*

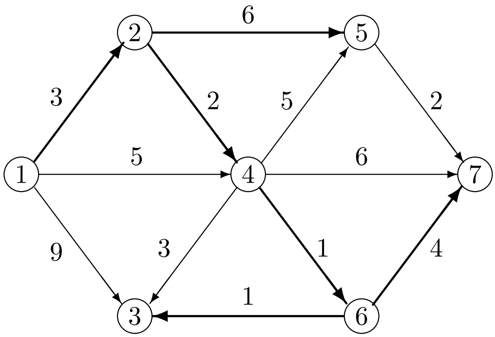


Рис. 6.2. Граф для примера 6.2

Работа алгоритма по итерациям представлена в табл. 6.2. Заметим, что множество S на итерации i состот из всех вершин из колонки w, которые находятся в строках $1, \ldots, i$. Дуги $(parent(v), v)$ дерева кратчайших путей изображены жирными линиями на рис. 6.2. □

Таблица 6.2

Итерации алгоритма Дейкстры

Ит.	w	d							$parent$						
		1	2	3	4	5	6	7	1	2	3	4	5	6	7
0		0	∞	∞	∞	∞	∞	∞	-	-	-	-	-	-	-
1	1	0	3	9	5	∞	∞	∞	-	1	1	1	-	-	-
2	2	0	3	9	5	9	∞	∞	-	1	1	1	2	-	-
3	4	0	3	8	5	9	6	11	-	1	4	1	2	4	4
4	6	0	3	7	5	9	6	10	-	1	6	1	2	4	6
5	3	0	3	7	5	9	6	10	-	1	6	1	2	4	6
6	5	0	3	7	5	9	6	10	-	1	6	1	2	4	6

6.1.5. Кратчайшие пути между всеми парами вершин

В графе $G = (V, E)$ с функцией стоимости $c : E \to \mathbb{R}$ нужно найти кратчайшие пути между всеми парами вершин. Эта задача имеет решение тогда и только тогда, когда в G нет циклов отрицательной стоимости. В таком случае, применяя алгоритм Форда — Беллмана к графу G_{aux}, введенному в доказательстве следствия 6.1, мы найдем функцию цен $p : V \to \mathbb{R}$, такую, что $c_p(v, w) \geq 0$ для всех $(v, w) \in E$. Применяя алгоритм Дейкстры к графу G с функцией стоимости c_p, для каждой вершины $s \in V$ мы найдем дерево кратчайших путей с корнем s. Временная сложность всей процедуры — $O(n^3)$.

6.2. Потоки и граф остаточных пропускных способностей

Потоковая сеть (G, l, u) — это орграф $G = (V, E)$, каждой дуге которого приписаны два числовых параметра $l(v, w)$ и $u(v, w)$, соответственно, ее *нижняя* и *верхняя пропускные способности*. Функция $f : E \to \mathbb{R}$ называется *псевдопотоком*, если выполняются следующие *ограничения на пропускные способности*:

$$l(v, w) \leq f(v, w) \leq u(v, w), \quad (v, w) \in E.$$

В теории удобнее рассматривать *симметричные потоковые сети* (G, u), где $G = (V, E)$ есть симметричный орграф, а $u : E \to \mathbb{R} \cup \{\infty\}$ — *функция пропускных способностей*. Напомним, что орграф $G = (V, E)$ называется симметричным, если для любой дуги $(v, w) \in E$ обратная дуга (w, v) также принадлежит E. Функция $f : E \to \mathbb{R}$ называется *псевдопотоком* в сети (G, u), если выполняются следующие условия:

$$f(v, w) \leq u(v, w) \quad \text{для всех } (v, w) \in E, \tag{6.3}$$

$$f(v, w) = -f(w, v) \quad \text{для всех } (v, w) \in E. \tag{6.4}$$

Условие (6.3) задает *ограничения на пропускные способности дуг*: по дуге не может течь поток, превышающий ее пропускную способность. *Условие антисимметрии потока* (6.4) отражает тот факт, что поток величины x из v в w есть поток величины $(-x)$ из w в v. Из условий (6.3) и (6.4) следует, что для псевдопотока всегда выпоняются неравества: $u(v, w) + u(w, v) \geq 0$ для всех дуг $(v, w) \in E$.

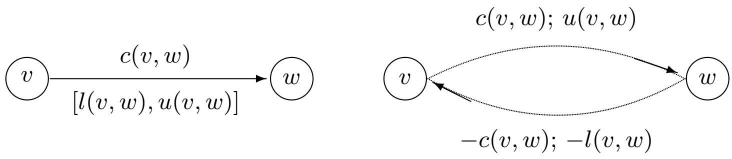


Рис. 6.3.

Заметим, что в симметричных сетях нет необходимости вводить нижние пропускные способности дуг, поскольку пропускная способность дуги (v,w) представляет нижнюю пропускную способность обратной дуги (w,v). Это позволяет преобразовать любую несимметричную потоковую сеть (G,l,u) в эквивалентную симметричную потоковую сеть $(\bar{G},\bar{u})$, где $\bar{G} \stackrel{\text{def}}{=} (V,\bar{E})$, $\bar{E} \stackrel{\text{def}}{=} E \cup \{(w,v) : (v,w) \in E\}$, а $\bar{u}(v,w) = u(v,w)$ и $\bar{u}(w,v) = -u(v,w)$ для всех $(v,w) \in E$. Иными словами, мы заменяем дугу (v,w) парой противоположных дуг (v,w), (w,v) и определяем их стоимости (во многих приложениях, также задается стоимость $c(v,w)$ транспортировки единицы потока по дуге (v,w)) и пропускные способности так, как показано на рис. 6.3.

Для псевдопотока f в симметричной потоковой сети (G,u) *остаточная пропускная способность* дуги $(v,w) \in E$ есть $u_f(v,w) \stackrel{\text{def}}{=} u(v,w) - f(v,w)$. *Граф остаточных пропускных способностей* $G_f \stackrel{\text{def}}{=} (V,E_f)$ для псевдопотока f имеет множество дуг $E_f \stackrel{\text{def}}{=} \{(v,w) \in E : u_f(v,w) > 0\}$.

Граф G_f остаточных пропускных способностей для псевдопотока f в несимметричной потоковой сети (G,l,u) — это граф $\bar{G}_f$ для эквивалентной симметричной сети $(\bar{G},\bar{u})$. Пример построения графа остаточных пропускных способностей псевдопотока в несимметричной сети представлен на рис. 6.4.

Альтернативно, для псевдопотока f в несимметричной потоковой сети (G,l,u) мы можем определить граф $G_f \stackrel{\text{def}}{=} (V,E_f)$ остаточных пропускных способностей следующим образом. Сначала определим *остаточную пропускную способность*

а) (прямой) дуги дуги $(v,w) \in E$ по правилу: $u_f(v,w) \stackrel{\text{def}}{=} u(v,w) - f(v,w)$;

б) обратной дуги (w,v) к дуге $(v,w) \in E$ по правилу: $u_f(w,v) \stackrel{\text{def}}{=} f(v,w) - l(v,w)$.

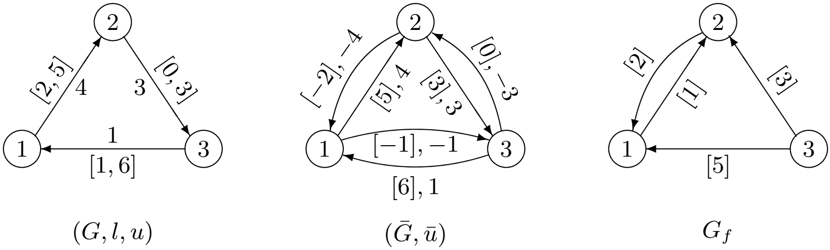


Рис. 6.4. Граф остаточных пропускных способностей

Теперь E_f задается следующим образом:

$$E_f \stackrel{\text{def}}{=} \{(v,w) \in E : u_f(v,w) > 0\} \cup \{(w,v) :\ (v,w) \in E,\ u_f(w,v) > 0\}.$$

В дальнейшем все теоретические результаты и описания алгоритмов приводятся для симметричных сетей. Чтобы использовать эти алгоритмы для решения соответствующих задач на несимметричных сетях, нам нужно только уметь строить графы остаточных пропускных способностей для несимметричных сетей, и помнить, что увеличение потока по обратной дуге (w,v) на ϵ в силу условия антисимметрии потока (6.4) влечет уменьшение потока на ϵ по прямой дуге (v,w).

6.3. Разложение потоков на элементарные

Для данного псевдопотока f в симметричной потоковой сети (G,u) определим *излишек* в вершине $v \in V$ как суммарный поток, втекающий в эту вершину:

$$e_f(v) \stackrel{\text{def}}{=} \sum_{(w,v)\in E} f(w,v)$$

Мы будем говорить, что вершина v имеет *излишек*, если $e_f(v) > 0$, и имеет *дефицит*, если $e_f(v) < 0$. Для вершины $v \in V$ *условие сохранения потока* задается равенством

$$e_f(v) = 0 \tag{6.5}$$

Циркуляцией называется псевдопоток с нулевым излишком в каждой вершине.

Лемма 6.6. *Пусть заданы два псевдопотока f и g, причем $e_f(v) > e_g(v)$. Тогда существуют вершина w с $e_f(w) < e_g(w)$ и последовательность различных вершин $v = v_0, v_1, \ldots, v_{l-1}, v_l = w$, такая, что $(v_{i-1}, v_i) \in E_f$ и $(v_i, v_{i-1}) \in E_g$ для $1 \le i \le l$.*

Доказательство. Определим орграфы $G_+ \stackrel{\text{def}}{=} (V, E_+)$ и $G_- \stackrel{\text{def}}{=} (V, E_-)$, где

$$\begin{aligned} E_+ &\stackrel{\text{def}}{=} \{(x,y) \in E : g(x,y) > f(x,y)\}, \\ E_- &\stackrel{\text{def}}{=} \{(x,y) \in E : f(x,y) > g(x,y)\}. \end{aligned}$$

Тогда $E_+ \subseteq E_f$, так как для $(x,y) \in E_+$ имеем $f(x,y) < g(x,y) \le u(x,y)$. Аналогично, $E_- \subseteq E_g$. Более того, по антисимметрии потока $(x,y) \in E_+$, если и только если $(y,x) \in E_-$. Поэтому достаточно показать, что в G_+ существует простой путь $v = v_0, v_1, \ldots, v_l = w$ з $e_f(w) < e_g(w)$.

Пусть S есть множество вершин, которые достигаются из вершины v в графе G_+, и пусть $\overline{S} = V \setminus S$ (множество $\overline{S}$ может быть пустым). Для дуги $(x,y) \in E(S, \overline{S})$ справедливо неравенство $f(x,y) \ge g(x,y)$, потому что в противном случае $y \in S$. Отсюда мы имеем

$$\begin{aligned} -\sum_{x \in S} e_g(x) &= \sum_{(x,y) \in E(S,V)} g(x,y) \\ &= \sum_{(x,y) \in E(S,\overline{S})} g(x,y) + \sum_{(x,y) \in E(S,S)} g(x,y) \\ &= \sum_{(x,y) \in E(S,\overline{S})} g(x,y) \quad \text{(по антисимметрии)} \\ &\le \sum_{(x,y) \in E(S,\overline{S})} f(x,y) \quad \text{(выполняется для каждого члена)} \\ &= \sum_{(x,y) \in E(S,\overline{S})} f(x,y) + \sum_{(x,y) \in E(S,S)} f(x,y) \quad \text{(по антисимметрии)} \\ &= \sum_{(x,y) \in E(S,V)} f(x,y) = -\sum_{x \in S} e_f(x). \end{aligned}$$

Так как $v \in S$ и $e_f(v) > e_g(v)$, то для некоторой вершины $w \in S$ должно выполняться неравенство $e_f(w) < e_g(w)$. □

Полезным свойством псевдопотоков является тот факт, что они могут быть разложены на небольшое количество *примитивных элемен-*

тов. Такими примитивными элементами являются элементарные потоки и циркуляции.

Псевдопоток g в сети G называется *элементарным потоком* (*циркуляцией*) величины ϵ, если подграф графа G с множеством дуг, на которых псевдопоток g положителен, является простым путем (циклом) и на всех дугах этого пути (цикла) величина потока равна ϵ.

Теорема 6.4. *Для любого псевдопотока f в сети G существует такое семейство элементарных потоков и циркуляций $f_1, \ldots, f_k$, $k \leq m$, что*

$$f(v,w) = \sum_{i=1}^{k} f_i(v,w) \quad \textit{для всех } (v,w) \in E. \tag{6.6}$$

Доказательство. Индукцией по количеству дуг с ненулевым потоком. Теорема справедлива для нулевого псевдопотока $f \equiv 0$. Пусть f — ненулевой псевдопоток. Допустим, что теорема верна для всех псевдопотоков f', таких, что $|E(f')| < |E(f)|$. Здесь $E(f) \stackrel{\text{def}}{=} \{(v,w) \in E : f(v,w) > 0\}$. Возможны два случая:

1) существует вершина $v \in V$ с положительным излишком $e_f(v) > 0$;
2) $e_f(v) = 0$ для всех $v \in V$ (f — циркуляция).

Случай 1. По лемме 6.6, примененной для $g \equiv 0$ и f, существует вершина w с $e_f(w) < 0$ и простой путь $w = v_0, v_1, \ldots, v_{l-1}, v_l = v$ в графе G_g, для которого $f(v_{i-1}, v_i) > 0$ для $i = 1, \ldots, l$. Пусть f_1 — элементарный поток, в котором по дугам этого пути течет поток величины

$$\epsilon = f(v_{j-1}, v_j) = \min_{1 \leq i \leq l} f(v_{i-1}, v_i).$$

Определим псевдопоток f' по правилу:

$$f'(x,y) = f(x,y) - f_1(x,y) \quad \text{для всех } (x,y) \in E.$$

Заметим, что $f'(v_{j-1}, v_j) = 0$. Поэтому $|E(f')| < |E(f)|$ и по индукционному допущению псевдопоток f' может быть представлен как сумма элементарных потоков:

$$f'(x,y) = \sum_{i=2}^{k} f_i(x,y) \quad \text{для всех } (x,y) \in E,$$

где $k \leq m$. Поэтому

$$f(x,y) = \sum_{i=1}^{k} f_i(x,y) \quad \text{для всех } (x,y) \in E.$$

Случай 2. Так как $f \not\equiv 0$, то существует дуга $(v,w) \in E$ с ненулевым потоком $f(v,w)$. Для конкретности будем считать, что $f(v,w) = -f(w,v) > 0$ (иначе нужно рассматривать дугу (w,v)). Построим псевдопоток $\overline{f}$ по правилу:

$$\overline{f}(v,w) = 0, \quad \overline{f}(x,y) = f(x,y) \quad \text{для всех } (x,y) \in E \setminus (v,w).$$

Понятно, что $e_{\overline{f}}(v) = -e_{\overline{f}}(w) = f(v,w) > 0$. Поэтому по лемме 6.6, примененной для $g \equiv 0$ и $\overline{f}$, существует вершина w с $e_{\overline{f}}(w) < 0$ и простой путь $w = v_0, v_1, \ldots, v_{l-1}, v_l = v$ в графе G_g, для которого $\overline{f}(v_{i-1}, v_i) > 0$ для всех $1 \leq i \leq l$. Пусть f_1 — элементарная циркуляция, в которой по дугам цикла $w = v_0, v_1, \ldots, v_{l-1}, v_l, v_{l+1} = w$ течет поток величины

$$\epsilon = \min_{1 \leq i \leq l+1} f(v_{i-1}, v_i) > 0.$$

Далее доказательсто такое же, как и в случае 1. $\square$

6.4. Сетевая транспортная задача

Функция стоимости есть действительная функция на множестве дуг $c : E \to \mathbb{R}$. Без ограничения общности мы допускаем, что она *антисимметричная*:

$$c(v,w) = -c(w,v) \quad \text{для всех } (v,w) \in E. \tag{6.7}$$

Стоимость псевдопотока f определяется по формуле:

$$c(f) \stackrel{\text{def}}{=} \sum_{(v,w) \in E : f(v,w) \geq 0} c(v,w) f(v,w).$$

В дополнение к фукциям стоимости c и пропускных способностей дуг u также дана *функция спроса* $d : V \to \mathbb{R}$, такая, что $\sum_{v \in V} d(v) = 0$. Псевдопоток называем *допустимым*, если выполняется следующее *условие сохранения потока*:

$$e_f(v) = d(v) \quad \text{для всех } v \in V. \tag{6.8}$$

Транспортная задача есть оптимизационная задача, в которой в *транспортной сети* (G, u, c, d) нужно найти допустимый псевдопоток минимальной стоимости.

Замечание. Сетевую транспортную задачу в более общей постановке, когда нельзя пренебречь фиксированными издержками, мы рассматривали в разделе 4.6.1. При отсутствии фиксированных издержек (все $f_e = 0$), задача (4.16) превращается в задачу ЛП, если положить $y_e = 1$ для всех $e \in E$. Можно сказать, что в данном разделе мы изучаем более эфективные (чем общие методы ЛП) методы такой задачи ЛП.

Задача о циркуляции минимальной стоимости является специальным случаем транспортной задачи, если $d(v) = 0$ для всех $v \in V$. В свою очередь, транспортную задачу в сети (G, u, c, d) можно преобразовать в задачу о циркуляции минимальнаой стоимости. Для этого к G добавим новую вершину $s \notin V$ и множество дуг: $\{(s,v),(v,s) :\ v \in V,\ d(v) \neq 0\}$. Стоимости и пропускные способности доопределяются следующим образом:

- когда $d(v) > 0$, то
$$u(v,s) = d(v),\ c(v,s) = -\infty, \quad u(s,v) = 0,\ c(s,v) = \infty;$$
- когда $d(v) < 0$, то
$$u(s,v) = -d(v),\ c(s,v) = -\infty, \quad u(v,s) = 0,\ c(v,s) = \infty.$$

Нетрудно убедиться, что ограничение циркуляции в расширенной сети на дуги исходной сети будет допустимым решением транспортной задачи, при этом оптимальной циркуляции соответствует оптимальное допустимое решение транспортной задачи.

6.4.1. Критерии оптимальности

Критерии оптимальности являются той основой, на которой базируются алгоритмы решения оптимизационных задач. Начнем с простой леммы.

Лемма 6.7. *Пусть f и g — допустимые псевдопотоки в транспортной сети (G, u, c, d). Тогда $g - f$ есть допустимая циркуляция в сети (G, u_f, c).*

Доказательство. Поскольку
$$g(v,w) - f(v,w) \le u(v,w) - f(v,w) = u_f(v,w),$$
то ограничения на пропускные способности выполняются. А так как
$$e_{g-f}(v) = e_g(v) - e_f(v) = d(v) - d(v) = 0,$$

то какже выполняется условие сохранения потока во всех вершинах $v \in V$. $\square$

Теперь мы сформулируем два критерия оптимальности псевдопотока.

Теорема 6.5. *Допустимый псевдопоток f в транспортной сети (G, u, c, d) оптимален тогда и только тогда, когда его граф остатосных пропускных способностей G_f не имеет циклов отрицательной стоимости.*

Доказательство. **Необходимость**. Пусть в графе G_f есть цикл

$$\Gamma = (v_0, v_1, \dots, v_{l-1}, v_l = v_0)$$

отрицательной стоимости $c(\Gamma) < 0$. Для

$$\epsilon \stackrel{\text{def}}{=} \min_{1 \le i \le l} u_f(v_{i-1}, v_i) > 0$$

построим допустимый псевдопоток f' следующим образом:

$$\begin{aligned} f'(v_{i-1}, v_i) &= f(v_{i-1}, v_i) + \epsilon, \quad i = 1, \dots, l, \\ f'(v_i, v_{i-1}) &= -f'(v_{i-1}, v_i), \quad i = 1, \dots, l, \\ f'(v, w) &= f(v, w), \quad (v, w) \in E \setminus E(\Gamma). \end{aligned}$$

Так как $c(f') = c(f) + \epsilon\, c(\Gamma) < c(f)$, то f не является оптималеным.

Достаточность. Допустим, что граф G_f не имеет циклов отрицательной стоимости. Пусть h — оптимальный допустимый псевдопоток в сети (G, u, c, d). По лемме 6.7 псевдопоток $(h - f)$ является циркуляцией в сети (G, u_f, c), а по теореме 6.4 существует такое семейство элементарных циркуляций $g_1, \dots, g_k$ в сети (G, u_f, c), что

$$h(v, w) - f(v, w) = \sum_{i=1}^{k} g_i(v, w) \quad \text{для всех } (v, w) \in E,$$

Так как

$$\begin{aligned} 0 &\ge c(h) - c(f) \\ &= \frac{1}{2} \sum_{(v,w) \in E} c(v, w) h(v, w) - \frac{1}{2} \sum_{(v,w) \in E} c(v, w) f(v, w) \end{aligned}$$

$$= \frac{1}{2} \sum_{(v,w)\in E_f} c(v,w)(h(v,w) - f(v,w))$$
$$= \frac{1}{2} \sum_{(v,w)\in E_f} c(v,w) \sum_{i=1}^{k} g_i(v,w)$$
$$= \sum_{i=1}^{k} \sum_{(v,w)\in E} \frac{1}{2} c(v,w) g_i(v,w)$$
$$= \sum_{i=1}^{k} c(g_i) \geq 0.$$

Отсюда имеем, что f также есть оптимальный допустимый псевдопоток. □

Непосредственно из леммы 6.1 получаем следующее утверждение.

Лемма 6.8. *Допустимый псевдопоток f является оптимальным в транспортной сети (G, u, c, d) тогда и только тогда, когда он оптимален в сети (G, u, c_p, d) для любой функции цен p.*

Второй критерий оптимальности допустимого псевдопотока следующий.

Теорема 6.6. *Допустимый псевдопоток f является оптимальным в транспортной сети (G, u, c, d) тогда и только тогда, когда существует такая функция цен p, что выполняется условие дополняющей нежесткости*

$$c_p(v,w) \geq 0 \quad \text{для всех } (v,w) \in E_f. \tag{6.9}$$

Доказательство. По теореме 6.5 псевдопоток f оптимален тогда и только тогда, когда граф G_f не имеет циклов отрицательной стоимости. Согласно следствию 6.1 граф G_f не имеет циклов отрицательной стоимости тогда и только тогда, когда существует такая функция $p : V \to \mathbb{R}$, которая удовлетворяет условию (6.9). □

Транспортная задача может не иметь оптимального решения.

Теорема 6.7. *Транспортная задача на сети (G, u, c, d) имеет оптимальное решение тогда и только тогда, когда она имеет допустимое решение и граф G не имеет циклов отрицательной стоимости, у всех дуги которого имеют бесконечные пропускные способности.*

6.4.2. Сетевой симплекс-метод

Рассмотрим транспортную задачу на сети (G, u, c, d). Зафиксируем произвольную вершину $s \in V$. Пусть f — некоторый допустимый псевдопоток, T — ордерево с корнем s в графе G. В контексте сетевого симплекс-метода дерево T называется *базисным*. Вычислим функцию p расстояний дерева T. По определению функции расстояний $c_p(v, w) = 0$ для всех дуг $(v, w) \in E(T)$, а также и для всех обратных им дуг. Возможны два случая:

1) $c_p(v, w) \geq 0$ для всех дуг $(v, w) \in E_f$;
2) существует дуга $(v, w) \in E_f$ такая, что $c_p(v, w) < 0$.

В первом случае по теореме 6.8 поток f является оптимальным. Рассмотрим второй случай. Пусть $P = (w = v_0, v_1, \ldots, v_k = v)$ есть единственный (неориентированный) путь в дереве T из w в v. Так как граф G симметричный, то P также является ориентированным путем в графе G. Его стоимость равна $p(v) - p(w)$. Тогда

$$p(v) - p(w) + c(v, w) = c_p(v, w) < 0$$

есть стоимость цикла $(v_0, v_1, \ldots, v_k, v_{k+1} = v_0)$. Пусть

$$\delta = \min\{u_f(v_i, v_{i+1}) : 0 \leq i \leq k\}.$$

Дуга $(x, y) = (v_j, v_{j+1})$ пути P, такая, что $u_f(x, y) = \delta$, называется *блокирующей*. Заметим, что значение δ может быть равно нулю. Изменим псевдопоток f по правилу:

$$\begin{aligned} f(v_i, v_{i+1}) &:= f(v_i, v_{i+1}) + \delta, \\ f(v_{i+1}, v_i) &:= -f(v_i, v_{i+1}), \quad i = 0, \ldots, k, \\ f(x, y) &:= f(x, y) \quad \text{для остальных дуг.} \end{aligned}$$

При этом стоимость псевдопотока уменьшится на $-\delta c_p(v, w)$. Такая стратегия улучшения допустимого псевдопотока лежит в основе *сетевого симплекс-метода*, известного также как *метод потенциалов*. Процедура *net_simplex*, которая реализует сетевой симплекс-метод, представлена в листинге 6.3.

Пример 6.3. *Сетевым симплекс-методом решить транспортную задачу на сети, которая приведена на рис. 6.5.*

Чтобы найти начальный допустимый псевдопоток, расширим сеть (см. рис. 6.6), добавляя дополнительную вершину 6 и 5 дуг: $(2, 6)$, $(4, 6)$,

Вход: транспортная сеть G, u, c, d и допустимый псевдопоток f.
Выход: оптимальные допустимый псевдопоток f и функция цен p.

1. Выбрать покрывающее ордерево T графа G с корнем $s \in V$.
2. Вычислить функцию p расстояний ордерева T.
3. Пока существует дуга $(v, w) \in E_f$, такая, что $c_p(v, w) < 0$, выполнять шаги 3.1–3.5:
   3.1. В графе $(V, E(T) \cup \{(v, w)\})$ выделить цикл $(v_0 = v, w = v_1, v_2, \ldots, v_k = v)$.
   3.2. Вычислить $\delta = \min\limits_{0 \le i < k} u_f(v_i, v_{i+1})$.
   3.3. Выбрать дугу $(x, y) = (v_j, v_{j+1})$, такую, что $\delta = u_f(v_j, v_{j+1})$.
   3.4. Если $((x, y) \neq (v, w))$, модифицировать ордерево T:
      3.4.1. Удалить из T дугу (x, y) и добавить дугу (v, w).
      3.4.2. Переориентировать дуги полученного дерева в направлении от корня s к листьям.
      3.4.3. Вычислить функцию p расстояний нового ордерева T.
   3.5. Для $i = 0, \ldots, k-1$ положить
      3.5.1 $f(v_i, v_{i+1}) := f(v_i, v_{i+1}) + \delta$;
      3.5.2. $f(v_{i+1}, v_i) := -f(v_i, v_{i+1})$.

Листинг 6.3. Сетевой симплекс-метод

$(5, 6)$ и $(6, 1)$, $(6, 3)$, по одной дуге для каждой вершины исходного графа. Стоимости всех новых дуг определим равными 10 (∞), а пропускные способности и потоки на дугах $(6, v)$ и $(v, 6)$ — равными $|d(v)|$. Решение начинаем с ордерева T, которое представлено на рис. 6.7) (числа на дугах — это их стоимости, а числа рядом с вершинами — это их цены). Ниже представлены итерации метода. Изменения дуговых потоков на итерациях отражены в табл. 6.3.

1. $c_p(1, 2) = -10 + 5 - 10 = -15 < 0$, $(v, w) = (1, 2)$, $C = (1, 2, 6, 1)$, $\delta = \min\{3, 1, 1\} = 1$, $(x, y) = (2, 6)$. Новое дерево и функция цен представлены на рис. 6.8 а.

2. $c_p(1, 3) = -10+1+10 = 1$, $c_p(2, 4) = -5+4-10 = -11$, $(v, w) = (2, 4)$, $C = (2, 4, 6, 1, 2)$, $\delta = \min\{3, 2, 4, 2\} = 2$, $(x, y) = (1, 2)$. Новое дерево и функция цен представлены на рис. 6.8 б.

3. $c_p(2, 1) = 6 - 5 + 10 = 11$, $c_p(1, 3) = -10 + 1 + 10 = 1$, $c_p(2, 5) = 6 + 1 - 10 = -3$, $(v, w) = (2, 5)$, $C = (2, 5, 6, 4, 2)$, $\delta = \min\{1, 3, 6, 2\} = 1$, $(x, y) = (2, 5)$. Данная итерация не меняет базисное дерево.

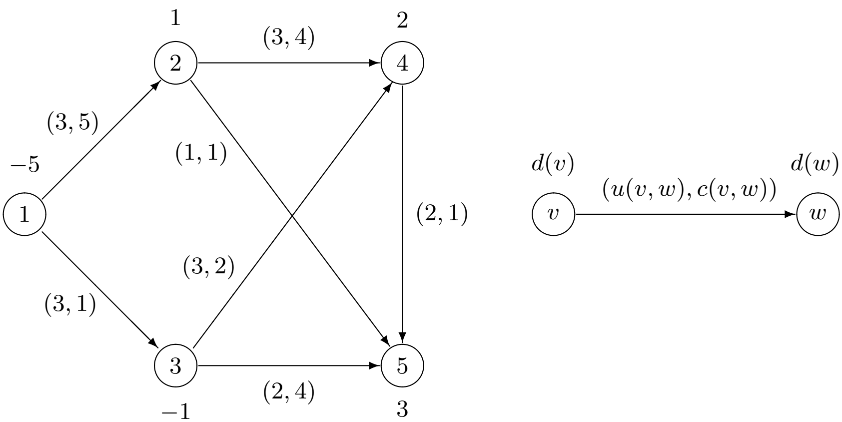


Рис. 6.5. Сеть для примера 6.3

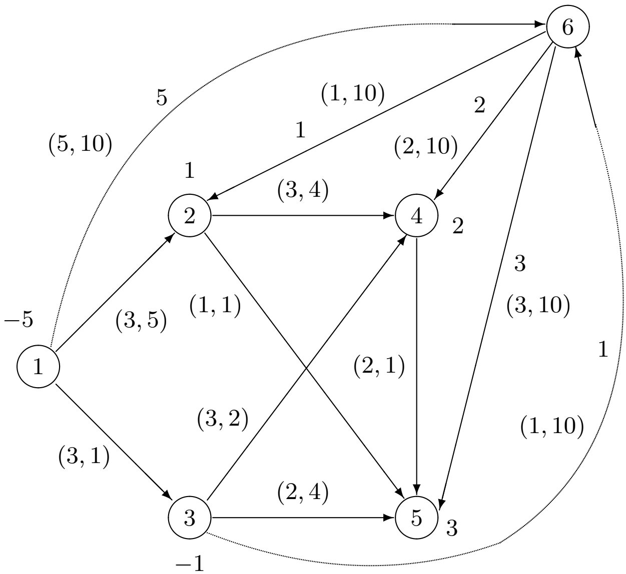


Рис. 6.6. Расширенная сеть для примера 6.3

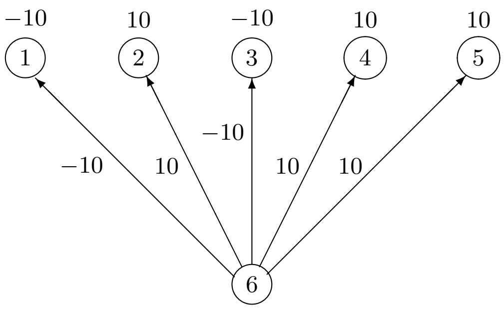


Рис. 6.7. Начальное базисное дерево для примера 6.3

Таблица 6.3

Псевдопотоки на разных итерациях

	(1,2)	(1,3)	(2,4)	(2,5)	(3,5)	(3,4)	(4,5)
0	0	0	0	0	0	0	0
1	1	0	0	0	0	0	0
2	3	0	2	0	0	0	0
3	3	0	1	1	0	0	0
4,5	3	0	1	1	0	1	0
6	3	2	1	1	2	1	0
7,8	2	3	0	1	2	2	0
9	2	3	0	1	3	1	0

	(1,6)	(6,2)	(3,6)	(6,4)	(6,5)
0	5	1	1	2	3
1	4	0	1	2	3
2	2	0	1	0	3
3	2	0	1	1	2
4,5	2	0	0	0	2
6	0	0	0	0	0
7,8	0	0	0	0	0
9	0	0	0	0	0

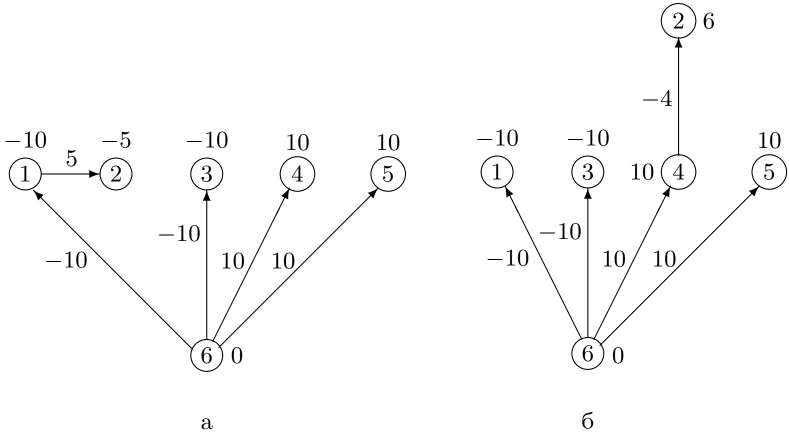


Рис. 6.8. Базисные деревья после итераций 1,2 и 3

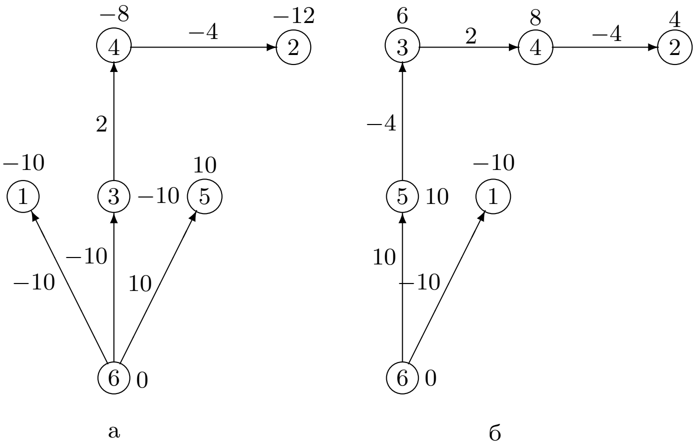
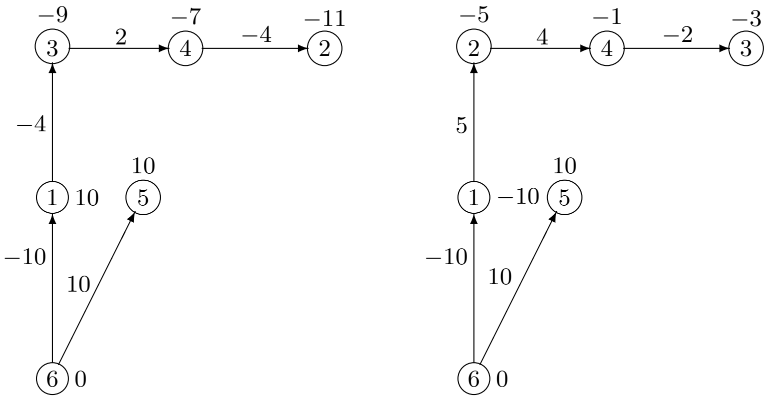
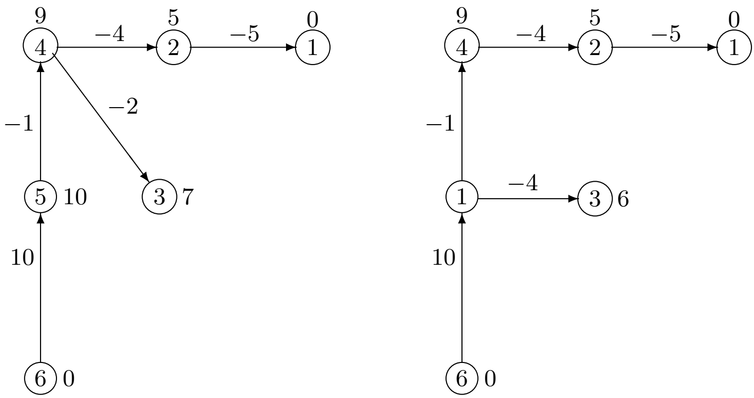


Рис. 6.9. Базисные деревья после итераций 4,5



а б

Рис. 6.10. Базисные деревья после итераций 6,7



а б

Рис. 6.11. Базисныя деревья после итераций 8,9

4. $c_p(3,4) = -10 + 2 - 10 = -18$, $(v,w) = (3,4)$, $C = (3,4,6,3)$, $\delta = \min\{3,1,1\}$, $(x,y) = (4,6)$. Новое дерево и функция цен представлены на рис. 6.9 а.

5. $c_p(3,5) = -10 + 4 - 10 = -16$, $(v,w) = (3,5)$, $C = (3,5,6,3)$, $\delta = \min\{2,2,0\} = 0$, $(x,y) = (6,3)$. Новое дерево и функция цен представлены на рис. 6.9 б.

6. $c_p(1,3) = -10 + 1 - 6 = -15$, $(v,w) = (1,3)$, $C = (1,3,5,6,1)$, $\delta = \min\{3,2,2,2\} = 2$, $(x,y) = (3,5)$. Новое дерево и функция цен представлены на рис. 6.10 а.

7. $c_p(2,1) = -11 - 5 + 10 = -6$, $(v,w) = (2,1)$, $C = (2,1,3,4,2)$, $\delta = \min\{3,1,2,1\} = 2$, $(x,y) = (1,3)$. Новое дерево и функция цен представлены на рис. 6.10 б.

8. $c_p(3,1) = -3-1+10 = 6$, $c_p(5,2) = 10-1+5 = 14$, $c_p(5,3) = 10-4+3 = 9$, $c_p(4,5) = -1+1-10 = -10$, $(v,w) = (4,5)$, $C = (4,5,6,1,2,4)$, $\delta = \min\{2,0,6,1,3\} = 0$, $(x,y) = (5,6)$. Новое дерево и функция цен представлены на рис. 6.11 а.

9. $c_p(3,1) = -3 - 1 + 10 = 6$, $c_p(5,2) = 10 - 1 - 5 = 4$, $c_p(5,3) = 10 - 4 - 7 = -1$, $(v,w) = (5,3)$, $C = (3,5,4,3)$, $\delta = \min\{2,1,2\} = 1$, $(x,y) = (3,4)$. Новое дерево и функция цен представлены на рис. 6.11 б.

$\square$

6.5. Задача о максимальном потоке

Дана сеть (G,u), в которой выделены две вершины $s,t \in V$; s называется *источником*, а t —*стоком*. *Предпотоком* называется псевдопоток f с неотрицательным излишком во всех вершинах, за исключением s. *Потоком* f в сети G называется псевдопоток, который удовлетворяет условию сохранения потока (6.5) во всех вершинах, кроме s и t. *Величиной потока* f называется количество потока, который втекает в сток $|f| = e_f(t)$. *Максимальный поток* — это поток максимальной величины. *Задача о максимальном потоке* есть задача поиска потока максимальной величины в заданной *потоковой сети* (G,u,s,t).

6.5.1. Критерии оптимальности

Используя теорему 6.5, получим два критерия оптимальности потока.

Теорема 6.8. *Поток f является максимальным потоком в сети (G, u, s, t) тогда и только тогда, когда в графе G_f отсутствуют пути из s в t.*

Доказательство. К потоковой сети (G, u, s, t) добавим дуги (t, s) и (s, t) с пропускной способностью $u(t, s) = \infty$ и $u(s, t) = 0$. Стоимости $c(v, w)$ всех дуг $(v, w) \in E$ определим равными нулю, а $c(t, s) = -1$, $c(s, t) = 1$. Обозначим расширенный граф через G^{st}.

Полагая $f(t, s) = e_f(t)$ и $f(s, t) = -f(t, s)$, доопределим поток f до циркуляции в сети (G^{st}, u, c). По теореме 6.5 циркуляция f оптимальна тогда и только тогда, когда в графе G_f^{st} нет циклов отрицательной стоимости. По построению сети (G^{st}, u, c) граф G_f^{st} имеет цикл отрицательной стоимости тогда и только тогда, когда в G_f есть путь из s в t. □

Пусть $S \subseteq V$. Напомним, что непустое множество $E(S, V \setminus S)$ называется *разрезом*. Разрез $E(S, V \setminus S)$ называется *s, t-разрезом*, если S является *$s, \bar{t}$-множеством*, т. е. $s \in S$, $t \notin S$. *Величиной разреза* называется сумма пропускных способностей его дуг:

$$\delta_u(S) = \sum_{(v,w) \in E(S, V \setminus S)} u(v, w).$$

Фундаментальной теоремой о потоках в сетях является следующая теорема.

Теорема 6.9 (Форд — Фалкерсон). *Величина максимального потока в потоковой сети (G, u, s, t) равна величине минимального s, t-разреза.*

Доказательство. Пусть f — произвольный поток в сети (G, u, s, t), а $E(S, V \setminus S)$ — произвольный s, t-разрез. Сложив равенства

$$\begin{aligned} |f| &= -e_f(s), \\ 0 &= -e_f(v), \quad v \in S \setminus s, \end{aligned}$$

получим

$$|f| = -\sum_{v \in S} e_f(v) = -\sum_{v \in S} \sum_{(w,v) \in E} f(w, v)$$

$$
\begin{aligned}
&= -\sum_{(w,v)\in E(V,S)} f(w,v) = \sum_{(v,w)\in E(S,V)} f(v,w) \\
&= \sum_{(v,w)\in E(S,V\setminus S)} f(v,w) + \sum_{(v,w)\in E(S,S)} f(v,w) \\
&= \sum_{(v,w)\in E(S,V\setminus S)} f(v,w) \le \sum_{(v,w)\in E(S,V\setminus S)} u(v,w) \\
&= \delta_u(S).
\end{aligned}
\tag{6.10}
$$

Пусть теперь f — максимальный поток, а S есть множество вершин графа G_f, достигаемых из источника s. Очевидно, что $s \in S$, а по теореме 6.8 $t \notin S$. Кроме того, $u(v,w) = f(v,w)$ для всех дуг $(v,w) \in E(S, V \setminus S)$. Но в этом случае неравенство (6.10) превращается в равенство и тогда $|f| = \delta_u(S)$. □

Следующий критерий существования циркуляции в сети (G,u) является следствием из теоремы Форда — Фалкерсона.

Теорема 6.10 (Гофман). *В сети (G,u) существует циркуляция тогда и только тогда, когда*

$$
\delta_u(X) \ge 0 \quad \text{для всех } X \subseteq V. \tag{6.11}
$$

Доказательство. **Необходимость**. Пусть в сети (G,u) существует циркуляция f. Тогда

$$
\begin{aligned}
\delta_u(X) &= \sum_{(v,w)\in E(X,V\setminus X)} u(v,w) \ge \sum_{(v,w)\in E(X,V\setminus X)} f(v,w) \\
&= \sum_{(v,w)\in E(X,V)} f(v,w) = -\sum_{v\in X} e_f(v) = 0.
\end{aligned}
$$

Достаточность. Для простоты мы обсудим случай, когда нет дуг с бесконечной пропускной способностью. (общий случай остается читателю). Построим псевдопоток f по правилу:

> если $u(v,w) \ge -u(w,v)$, то $f(v,w) = -u(w,v)$ и $f(w,v) = u(w,v)$.

Если $e_f(v) = 0$ для всех $v \in V$, то f — циркуляция. Иначе оба множества $S = \{v \in V : e_f(v) > 0\}$ и $T = \{v \in V : e_f(v) < 0\}$ не пустые. Построим орграф $G' = (V', E')$, где

$$
V' \stackrel{\text{def}}{=} V \cup \{s,t\}, \quad E' \stackrel{\text{def}}{=} E \cup \{(s,v) : v \in S\} \cup \{(v,t) : v \in T\}.
$$

Определим пропускные способности дуг следующим образом:

$$
\begin{aligned}
u'(s,v) &= e_f(v), \quad v \in S,\\
u'(v,t) &= -e_f(v), \quad v \in T,\\
u'(v,w) &= u(v,w) - f(v,w), \quad (v,w) \in E.
\end{aligned}
$$

Пусть $M = \sum_{v \in S} e_f(v)$. Нетрудно видеть, что в потоковой сети (G', u', s, t) существует поток x величины M тогда и только тогда, когда функция $f + x$ (ограниченная на E) является циркуляцией в (G, u). По теореме 6.9 величина максимального потока в (G', u', s, t) равна M тогда и только тогда, когда

$$
\delta_{u'}(X \cup \{s\}) \geq M \quad \text{для всех } X \subseteq V. \tag{6.12}
$$

Но

$$
\begin{aligned}
\delta_{u'}(X \cup \{s\}) &= \sum_{(v,w) \in E'(X \cup \{s\}, V \cup \{t\} \setminus X)} u'(v,w)\\
&= \sum_{(s,w): w \in S \setminus X} u'(s,w) + \sum_{(v,t): v \in X \cap T} u'(v,t) +\\
&\sum_{(v,w) \in E(X, V \setminus X)} u'(v,w)\\
&= \sum_{w \in S \setminus X} e_f(w) - \sum_{v \in X \cap T} e_f(v) +\\
&\sum_{(v,w) \in E(X, V \setminus X)} (u(v,w) - f(v,w))\\
&= \sum_{v \in S \setminus X} e_f(v) - \sum_{v \in X \cap T} e_f(v) +\\
&\delta_u(X) - \sum_{(v,w) \in E(X,V)} f(v,w)\\
&= \sum_{v \in S \setminus X} e_f(v) - \sum_{v \in X \cap T} e_f(v) + \delta_u(X) + \sum_{v \in X} e_f(v)\\
&= \sum_{v \in S \setminus X} e_f(v) - \sum_{v \in X \cap T} e_f(v) + \delta_u(X) +
\end{aligned}
$$

$$\begin{aligned}&\sum_{v\in X\cap S} e_f(v) + \sum_{v\in X\cap T} e_f(v)\\&= \sum_{v\in S} e_f(v) + \delta_u(X)\\&= M + \delta_u(X).\end{aligned}$$

Отсюда имеем, что условия (6.11) и (6.12) эквивалентны. □

В заключение сформулируем два комбинаторных следствия из теоремы Форда — Фалкерсона.

Теорема 6.11 (Менгер). *В орграфе $G = (V, E)$ имеется k путей из s в t, которые не имеют общих дуг, тогда и только тогда, когда $\rho(X) \geq k$ для всех $s, \bar{t}$-подмножеств $X \subseteq V$.*

Доказательство. Результат следует из теоремы Форда — Фалкерсона, примененной к потоковой сети (G, u, s, t), где пропускные способности всех дуг равны 1. □

Теорема 6.12 (Кениг). *Максимальная мощность паросочетания в двудольном графе $G = (V \cup W, E)$ равна минимальному количеству вершин, которые покрывают все ребра.*

Доказательство. Ориентируем ребра графа G в направлении от V к W. Затем добавим к G две новые вершины $s, t \notin V \cup W$ и два множества дуг

$$\{(s, v)\ :\ v \in V\}, \quad \{(v, t)\ :\ v \in W\}.$$

Пропускные способности u всех новых дуг равны 1, а дуг из E — ∞. Полученную потоковую сеть обозначим через (G^{+st}, u, s, t). Между паросочетаниями в графе G и целочисленными потоками в сети (G^{+st}, u, s, t) существует взаимно однозначное соответствие: ребро $(v, w) \in E$ принадлежит паросочетанию тогда и только тогда, когда $f(s, v) = f(v, w) = f(w, t) = 1$. Понятно, что максимальному потоку величины k соответствует паросочетание мощности k. По теореме 6.9 существует $s, \bar{t}$-множество X, для которого $\delta_u(X) = k$. Заметим, что ниодна дуга с бесконечной пропускной способностью не принадлежит $E(X, V \cup W \setminus X)$. Поэтому множество $(V \setminus X) \cup (W \cap X)$ имеет мощность k и покрывает все ребра. □

Полным паросочетанием в двудольном графе $G = (V \cup W, E)$ с $|V| = |W|$ называется паросочетание мощности $|V|$. Еще говорят, что полное паросочетание — это такое паросочетание, ребра которого покрывают все вершины. Каждое ребро покрывает только инцидентные ему вершины. Для $A \subseteq V$ обозначим через $W(A)$ множество вершин в W, смежных хотя бы одной вершине из A.

Теорема 6.13 (Холл). *В двудольном графе $G = (V \cup W, E)$ существует полное паросочетание тогда и только тогда, когда $|A| \leq |W(A)|$ для любого $A \subseteq V$.*

Доказательство этой теоремы мы оставляем читателю в качестве упражнения, а сами ограничимся следующей интерпретацией.

Во дворе короля Артура проживали 500 незамужних фрейлин и 500 холостых рыцарей. Король Артур решил всех их поженить и приказал своему магу Мерлину составить 500 пар так, чтобы в каждой паре как фрейлин так и рыцарь были согласны вступить в брак друг с другом. Обладая сверхспособностью в вычислениях, Мерлин, перебрав все возможные варианты решения поставленной перед ним задачи, пришел к выводу, что она не имеет решения, и сообщил об этом королю. Но короля такой ответ не устроил, и он приказал магу убедительно объяснить причину, почему нельзя сделать так, как он хочет. В противном случае король пригрозил Мерлину, что тот проведет остаток своих дней в заточении.

Нельзя с уверенностью сказать, догадался ли Мерлин сам, или перенесся во времени в 20-й век, чтобы найти ответ в книге по теории графов, но он все же убедил короля в невозможности составить пары из симпатизирующих друг другу фрейлин и рыцарей. Мерлин вызвал 59 рыцарей и затем попросил выйти вперед всех фрейлин, которые готовы выйти замуж за хотя бы одного из этих 59 рыцарей. Поскольку вперед вышли только 58 фрейлин, то королю стало ясно, что его задача действительно не имеет решения.

6.5.2. Алгоритм пометок

Для решения задачи о максимальном потоке из теоремы 6.8 непосредственно получаем алгоритм пометок Форда — Фалкерсона, который представлен в листинге 6.4.

Алгоритм пометок оставляет некоторую свободу в выборе пути для увеличения потока. В потоковой сети на рис. 6.12 увеличение потока вполне могло выполняться вдоль следующей последовательности путей: $(s, 1, 2, t)$,

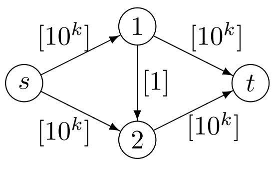


Рис. 6.12.

Вход: потоковая сеть $(G, u, s, t,)$ и начальный допустимый поток f.
Выход: оптимальный поток f и минимальный разрез $E(S, V \setminus S)$.
1. Пока в графе G_f существует путь $P = (s = v_0, v_1, \ldots, v_k = t)$:
   1.1. Вычислить минимальную остаточную пропускную способность дуг пути P: $\epsilon = \min\limits_{1<i\leq k} u_f(v_{i-1}, v_i)$.
   1.2. Изменить поток вдоль дуг пути P:
   $f(v_{i-1}, v_i) := f(v_{i-1}, v_i) + \epsilon$, $f(v_i, v_{i-1}) = -f(v_{i-1}, v_i)$, $i = 1, \ldots, k$.
2. Найти в G_f множество вершин S, достижимых из источника s.
3. Вернуть ответ $(f, E(S, V \setminus S))$.

Листинг 6.4. Алгоритм пометок

$(s, 2, 1, t)$, $(s, 1, 2, t)$, $(s, 2, 1, t)$ и т. д. Всего было бы выполнено 10^{2k} увеличений потока, причем, каждый раз всего на $\epsilon = 1$. Но если бы поток увеличивался вдоль кратчайших (по количеству дуг) путей, то алгоритму пометок потребовалось бы всего две итерации: на первой из которых поток увеличается вдоль пути $(s, 1, t)$ на $\epsilon = 10^k$, а второй — вдоль пути $(s, 2, t)$ на $\epsilon = 10^k$.

Диниц, а затем Эдмондс и Карп, предложили модификацию алгоритма, в которой увеличение потока проводится вдоль путей из s в t кратчайшей длины. Эту модификацию мы называем алгоритмом Эдмондса — Карпа (алгоритм Диница основан на этой и иных идеях и является более эффективным). Мы получим эту модификацию, если в графе G_f путь из s в t будем искать процедурой *поиска в ширину*, которая поддерживат список Q уже посещенных вершин в виде очереди. Это означает, что новые вершины всегда добавляются в конец списка, а выбор вершины из списка осуществляется с его начала. Как мы увидим позже, сложность алгоритма Эдмондса — Карпа полиномиально зависит от n и m.

Пример 6.4. *Найдем максимальный поток в сети, которая представлена на рис. 6.13.*

Начинаем решать задачу с нулевого потока. Ниже приведены итерации алгоритма пометок.

1. $Q = (s, 1, 2, 3, 4, t)$, $parent = (s, s, s, 1, 1, 3)$. Увеличиваем поток вдоль пути $(s, 1, 3, t)$ на $\epsilon = \min\{3, 4, 3\} = 3$.

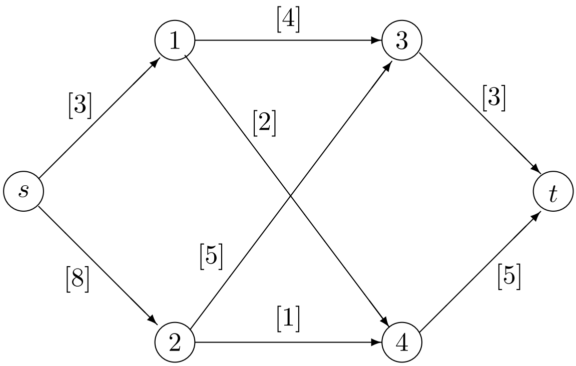


Рис. 6.13. Сеть для примера 6.4

2. $Q = (s, 2, 3, 4, 1, t)$, $parent = (s, 3, s, 2, 2, 4)$. Увеличиваем поток вдоль пути $(s, 2, 4, t)$ на $\epsilon = \min\{8, 1, 5\} = 1$.
3. $Q = (s, 2, 3, 1, 4, t)$, $parent = (s, 3, s, 2, 1, 4)$. Увеличиваем поток вдоль пути $(s, 2, 3, 1, 4, t)$ на $\epsilon = \min\{7, 2, 3, 2, 4\} = 2$.
4. $Q = (s, 2, 3, 1)$, $parent = (s, 3, s, 2, -, -)$. Поскольку в результате поиска в ширину мы не достигли стока t, то в графе G_f нет путей из s в t. Поэтому текущий поток величины 6 максимален, а множество дуг $E(\{1, 2, 3\}, \{4.t\}) = \{(1, 4), (2, 4), (3, t)\}$ есть минимальный $s, \bar{t}$-разрез величины $\delta_u(S) = 2 + 1 + 3 = 6$.

Изменения потока на итерациях алгоритма представлены в табл. 6.4.
□

Таблица 6.4

Итерации алгоритма пометок при решении примера 6.4

Ит.	f							
	$(s, 1)$	$(s, 2)$	$(1, 3)$	$(1, 4)$	$(2, 3)$	$(2, 4)$	$(3, t)$	$(4, t)$
0	0	0	0	0	0	0	0	0
1	3	0	3	0	0	0	3	0
2	3	1	3	0	0	1	3	1
3	3	3	1	2	2	1	3	3

Сложность алгоритма Эдмондса — Карпа

Пусть $\sigma_f(v)$ есть расстояние (длина кратчайшего пути) от s до v в графе G_f (если нет пути из s в v, то $\sigma_f(v) = \infty$. В графе G_f рассмотрим путь P кратчайшей длины из s в t. Ясно, что

$$\sigma_f(w) = \sigma_f(v) + 1 \quad \text{для каждой дуги } (v, w) \text{ пути } P. \tag{6.13}$$

Лемма 6.9. *Пусть f и g соответственно поток в начале и конце некоторой итерации алгоритма Эдмондса — Карпа. Тогда $\sigma_f(v) \leq \sigma_g(v)$ для всех $v \in V$.*

Доказательство. Рассмотрим, как изменится граф остаточных пропускных способностей G_f после увеличения потока f вдоль кратчайшего пути P из s в t на величину $\epsilon = \min\{u_f(v, w) : (v, w) \in P\}$. Так как поток меняется только на дугах пути P (и обратных к ним), то и граф остаточных пропускных способностей также изменится только на этих дугах: в графе G_g могут появиться новые дуги, которые являются обратными дугам пути P, и исчезнуть *критические* дуги (те, для которых $u_f(v, w) = \epsilon$) пути P. Расстояние от s до v может уменьшиться только тогда, когда добавится дуга (x, y), для которой $\sigma_f(y) > \sigma_f(x) + 1$, что невозможно по (6.13). □

Лемма 6.10. *На протяжении всего времени выполнения алгоритма Эдмондса — Карпа дуга (v, w) может быть критической не более чем n раз.*

Доказательство. При увеличении потока f по критической дуге (v, w) она исчезнет из G_f. Допустим, что позже при новом увеличении текущего потока g дуга (v, w) снова появится в графе остаточных пропускных способностей. Тогда $\sigma_g(v) = \sigma_g(w) + 1$. По лемме 6.9 мы имеем

$$\sigma_f(v) + \sigma_f(w) = 2\sigma_f(w) - 1 \leq 2\sigma_g(w) - 1 = \sigma_g(v) + \sigma_g(w) - 2.$$

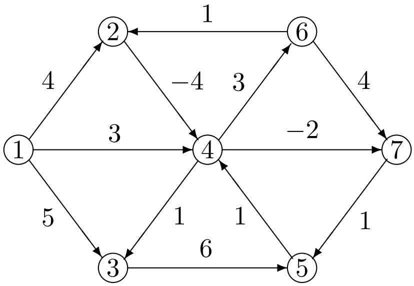
Так как расстояние от s до v не превосходит $n - 1$, то, очевидно, что лемма справедлива. □

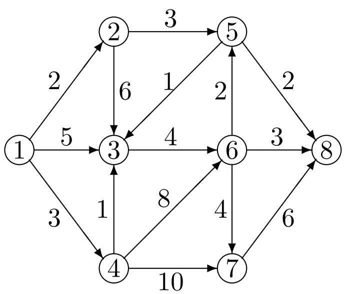
Теорема 6.14. *Сложность алгоритма Эдмондса — Карпа — $O(nm^2)$.*

Доказательство. По лемме 6.10 алгоритм увеличивает поток не более nm раз. Каждое увеличение может быть выполнено за время $O(m)$ ($O(m)$ требуется для нахождения пути P и $O(n)$ для изменения потока вдоль пути P). Поэтому общая сложность алгоритма — $O(nm^2)$. □

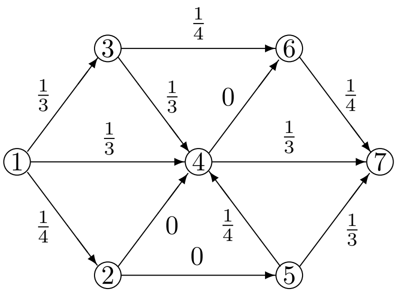
6.6. Упражнения

6.1. В графах, представленных на следующем рисунке, найти кратчайшие пути от вершины 1 до всех остальных вершин.

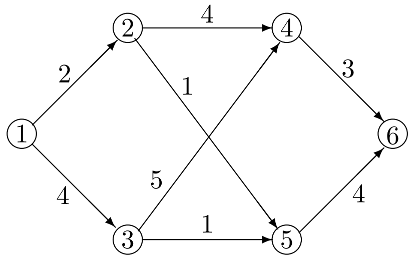




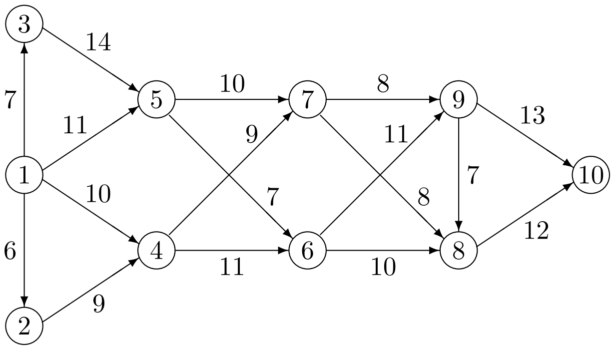
6.2. В коммуникационной сети, схема которой изображена на следующем рисунке, канал связи (дуга) (v, w) может засбоить с вероятностью $p(v, w)$. Числа $p(v, w)$ проставлены на дугах сети. Нужно выбрать наиболее надежный путь для пересылки сообщения из узла 1 в узел 7. Из двух путей более *надежен* тот путь, для которого вероятность того, что ни один из его каналов не засбоит, большая.



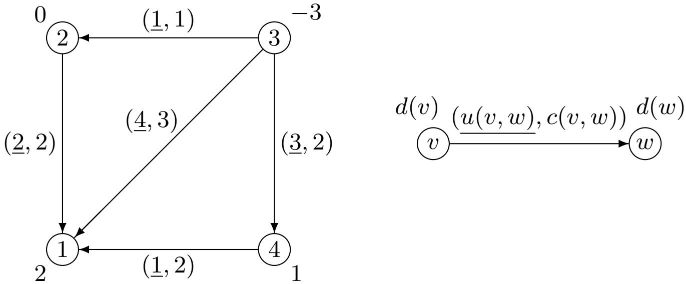
6.3. Найти максимальный поток из вершины $s = 1$ в вершину $t = 6$ в сети, которая представлена на следующем рисунке. Числа на дугах — это их (верхние) пропускные способности.



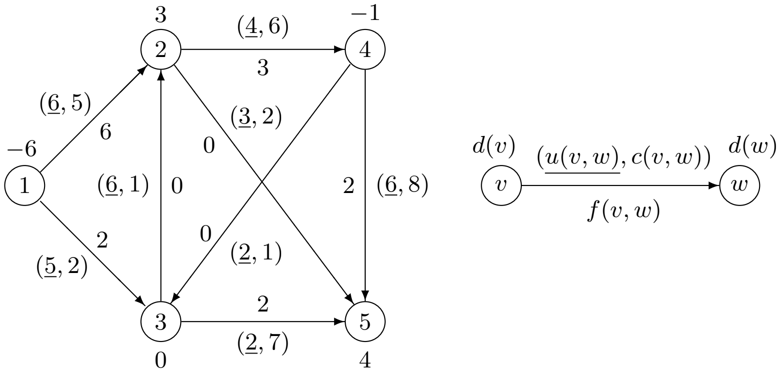
6.4. В потоковой сети, изображенной ниже, найти максимальной поток из вершины 1 в вершину 10. Числа на дугах — это их (верхние) пропускные способности.



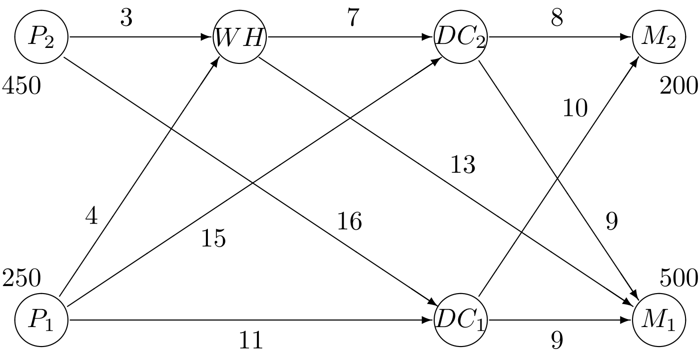
6.5. Решите транспортную задачу в следующей сети:



6.6. Является ли оптимальным псевдопоток, отмеченный на дугах следующей сети?



6.7. На рисунке



представлена цепочка поставок, в которой некоторый продукт, производимый на двух предприятиях P_1 и P_2, затем поставляется либо на склад WH, либо на два дистрибьютерских центра DC_1 и DC_2. На рынки M_1 и M_2 продукт поставляется со склада или из дистрибьютерских центров.

Производственный план такой, что предприятие P_1 производит 450 единиц продукта, а предприятие P_2 — 250 единиц. Спрос на рынке M_1 — 500 единиц продукта, а на рынке M_2 — 200 единиц. Числа на дугах — это стоимости транспортировки единицы продукта между соответствующими узлами. Пропускные способности всех дуг неограничены.

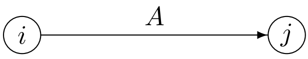
Нужно найти оптимальный план поставки продукта от производителей к потребителям. Как изменится этот план, если предположить, что емкость склада позволяет хранить не более 500 единиц продукта, а в каждом из дистрибьютерских центров можно хранить не более 200 единиц продукта?

Глава 7
Календарное планирование

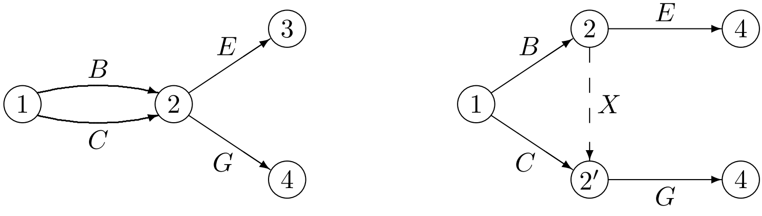
Управление крупными проектами связано с решением сложных проблем планирования, определения сроков начала и окончания отдельных работ, контроля за выполнением этих сроков. Все это осложняется тем, что работы должны выполняться в заданной технологической последовательности.

7.1. Сетевые графики

Сетевой график есть орграф $G = (V, E)$, дуги которого соответвуют работам, а сам граф отражает связи всеми заданиями, необходимыми для окончания проекта. Дуги графа являются *работами*, а вершины — событиями. *Событие* — это момент времени, когда можно начать выполнение новых работ. Для каждой работы (дуги) (i, j) известна ее продолжительность. Направление дуг определяется *отношениями предшествования.* На отрезке сети



i-е событие доджно наступить до начала работы A, а j-е событие не может наступить до окончания работы A. Иногда отношения предшествования между работами нельзя точно задать с помощью сети. Например, если работа G выполняется за работами B и C, а работа E — за работой B, но не за C. Приведенное на рис. 7.1, а представление отношений предшествования ошибочное, поскольку из него следует, что



а б

Рис. 7.1. Пример отношения предшествования, когда нужно вводить фиктивные работы

работа E следует и за работой C, а это не требуется. Для правильного представления нужно ввести *фиктивную* работу X продолжительности 0 (см. рис. 7.1, б). Фиктивные работы мы будем рисовать пунктирными линиями.

Пример 7.1. *При сборке некоторого станка узлы* 1 *и* 2 *соединяются в узел* 4, *а объединение узлов* 3 *и* 4 *дает готовое изделие. Так как необходимо согласовать некоторые детали узла* 3 *с соответствующими деталями узла* 2, *то узел* 3 *нельзя собрать ранее, чем будут в наличии детали узла* 2. *Основные работы проекта приведены в табл. 7.1.*

Сетевой график процесса изготовления станка представлен на рис. 7.2. □

Сетевой график проекта удовлетворяет следующим условиям.

1. Имеется одно начальное событие (вершина, в которую не входит ни одна дуга) и одно *заключительное* событие (вершина, и которой не выходит ни одна дуга).

2. В графике нет циклов. Поэтому события можна занумеровать таким образом, что каждая дуга (работа) начинается в вершине с меньшим номером и заканчивается в вершине с большим номером. В дальнейшем будем считать, что $V = \{1, \dots, n\}$ и, если $(i, j) \in E$, то $i < j$.

Нумерацию вершин можно выполнить, например, таким образом. Находим вершину, в которую не входит ни одна дуга и приписываем ей номер 1. Мысленно удаляем эту вершину и инцидентные ей дуги из

Таблица 7.1

Список работ для проекта из примера 7.1

Работа		Прод. (сут.)	Непоср. предш.
Обозн.	Описание		
A	Закупка деталей узла 1	5	–
B	Закупка деталей узла 2	3	–
C	Закупка деталей узла 3	10	–
D	Изготовление узла 1	7	A
E	Изготовление узла 2	10	B
F	Изготовление узла 4	5	D, E
G	Изготовление узла 3	9	B, C
H	Окончательная сборка	4	F, G
I	Испытания	2	H

графика. В оставшемся подграфе снова находим вершину, в которую не входит ни одна дуга и приписываем ей номер 2. Процесс повторяем, пока все вершины не получат свой номер. Такая нумерация вершин ациклического графа называется *топологической сортировкой.*

7.2. Метод критического пути

Одной из главных целей сетевого планирования является получение информации о плановых сроках выполнения отдельных работ проекта, что позволяет предвидеть возможные причины задержек.

7.2.1. Ранние и поздние сроки наступления событий

Ранний срок $T_j^{\text{р}}$ *наступления события* j есть ранний срок окончания всех работ, которые лежат на путях между начальным событием 1 и событием j. Таким образом, $T_j^{\text{р}}$ есть максимальная длина пути из вяршыны 1 в вершину j, если за длину дуги взять продолжительности работ. Параметры $T_j^{\text{р}}$ можно вычислить по следующей рекурентной

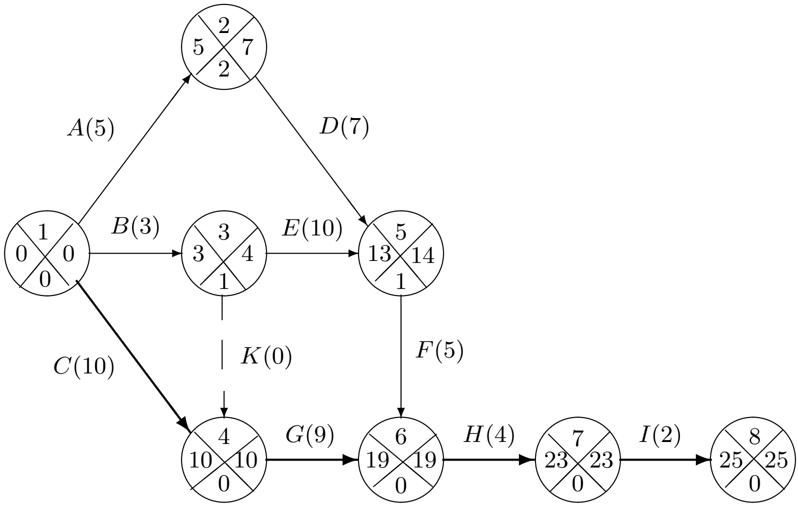


Рис. 7.2. Сетевой график процесса изготовления станка

формуле:

$$T_1^{\text{р}} = 0,$$
$$T_j^{\text{р}} = \max_{(i,j)\in E} \{T_i^{\text{р}} + t_{ij}\}, \quad j = 2, \dots, n.$$

В сетевом графике на рис. 7.2 ранние сроки наступления событий представлены в левых секторах.

Ранний срок наступления последнего события n равен раннему сроку окончания всего проекта. Таким образом, ранний срок окончания всего проекта равен максимальной длине пути из начального события 1 до заключительного события n. Этот путь называется *критический путем*, а его длина —*критическим временем*, которое обозначим через $T^{\text{кр}}$. На рис. 7.2 дуги критического пути нарисованы двумя стрелками.

Поздний срок $T_j^{\text{п}}$ наступления события j — это наиболее поздний срок наступления события j, который не влияет на ранний срок окончания всего проекта в целом (критическое время). Чтобы не увеличить ранний срок окончания проекта, j-е событие должно наступить не позже, чем в момент

$$T_j^{\text{п}} = T^{\text{кр}} - L_{jn},$$

где L_{jn} — максимальная длина пути из j в n. Мы можем вычислить параметры $T_j^{\text{п}}$ по следующей рекурентной формуле:

$$T_n^{\text{п}} = T^{\text{кр}},$$
$$T_j^{\text{п}} = \min_{(j,i)\in E}\{T_i^{\text{р}} - t_{ij}\}, \quad j = n-1, \dots, 1.$$

В сетевом графике на рис. 7.2 поздние сроки наступления событий представлены в правых секторах.

Резерв времени R_j события j — это максимальное время, на которое можно задержать наступление события без увеличения раннего срока окончания проекта, т.е.

$$R_j = T_j^{\text{п}} - T_j^{\text{р}}.$$

Событие с нулевым резервом времени находится на критическом пути. Задержка наступления любого события на критическом пути приводит к задержке всего проекта. Наоборот, наступление события j, которое не лежит на критическом пути может быть задержано на R_j единиц времени, причем, это не приведет к увеличению раннего срока окончания всего проекта. На рис. 7.2 резервы времени событий представлены в нижних секторах.

7.2.2. Ранние и поздние сроки начала и окончания работ

1. *Ранний срок* $T_{\text{н}}^{\text{р}}(i,j)$ *начала работы* (i,j) равен раннему сроку $T_i^{\text{р}}$ наступления события i, поскольку работа (i,j) не может быть начата, пока не наступит событие i.

2. *Поздний срок* $T_{\text{о}}^{\text{п}}(i,j)$ *окончания работы* (i,j) — это наиболее поздний срок окончания работы (i,j) без задержки срока окончания проекта: $T_{\text{н}}^{\text{р}}(i,j) = T_j^{\text{п}}$.

3. *Ранний срок* $T_{\text{о}}^{\text{р}}(i,j)$ *окончания работы* (i,j) определяется формулой $T_{\text{о}}^{\text{р}}(i,j) = T_j^{\text{р}} + t_{ij}$.

3. *Поздний срок* $T_{\text{н}}^{\text{п}}(i,j)$ *начала работы* (i,j) определяется формулой $T_{\text{н}}^{\text{п}}(i,j) = T_j^{\text{п}} - t_{ij}$.

7.2.3. Четыре показателя резерва времени работы

Эти показатели могут быть использованы руководителем проекта при распределении ресурсов (например, рабочей силы) для выполнения отдельных работ проекта, так как продолжительность работы зависит от количества выделенных ресурсов.

1. *Суммарный резерв* $R^{\text{сум}}(i, j)$ времени работы (i, j) — это максимальная задержка работы (i, j) без задержки срока выполнения всего проекта:

$$R^{\text{сум}}(i, j) = T_j^{\text{п}} - T_i^{\text{р}} - t_{ij}.$$

Для работ на критическом пути $R^{\text{сум}} = 0$. Если работа (i, j) целиком использует суммарный резерв, то в графике появится новый критический путь, который проходит через дугу (i, j).

2. *Свободный резерв* $R^{\text{св}}(i, j)$ времени работы (i, j) — это максимальная задержка работы (i, j), которая не влияет на начало последующих работ, т. е. последующие работы могут начинаться в свои ранние сроки:

$$R^{\text{св}}(i, j) = T_j^{\text{р}} - T_i^{\text{р}} - t_{ij}.$$

3. *Гарантированный резерв* $R^{\text{гар}}(i, j)$ времени работы (i, j) — это максимальная возможная задержка работы (i, j), которая не влияет на ранний срок окончания всего проекта, при условии что предшествующие работы выполнялись с опоздаеием в свои поздние сроки.

$$R^{\text{гар}}(i, j) = T_j^{\text{п}} - (T_i^{\text{п}} + t_{ij})$$

4. *Независимый резерв* $R^{\text{нез}}(i, j)$ времени работы (i, j) — это такая задержка работы (i, j), которая не влияет на начало следующих работ, при условии что все предшествующие работы окончились в свои поздние сроки:

$$R^{\text{нез}}(i, j) = \max\{0, T_j^{\text{р}} - T_i^{\text{п}} - t_{ij}\}$$

Продолжение примера 7.1. Результаты вычислений по методу критического пути представлены в табл. 7.2. В последнем столбце этой таблицы резервы времени работ представлены в следуюшем порядке: $(R^{\text{сум}}, R^{\text{св}}, R^{\text{нез}}, R^{\text{гар}})$.

По данным из табл. 7.2 можно нарисовать *временную диаграмму проекта* на координатной плоскости, откладывая по оси Ox — время, а по оси Oy — работы. Временная диаграмма для примера 7.1 приведена на рис. 7.3. □

Таблица 7.2

Параметры сетевого графика на рис. 7.2

Раб.	Дуга	Прод.	Ранний срок		Поздний срок		Резервы
			Нач.	Ок.	Нач.	Ок.	
A	$(1,2)$	5	0	5	2	7	$(2,0,0,2)$
B	$(1,3)$	3	0	3	1	4	$(1,0,0,4)$
C	$(1,4)$	10	0	10	0	10	$(0,0,0,0)$
D	$(1,5)$	7	5	12	7	14	$(2,1,0,0)$
E	$(3,5)$	10	3	13	4	14	$(1,0,0,0)$
G	$(4,6)$	9	10	19	10	19	$(0,0,0,0)$
F	$(5,6)$	5	13	19	14	19	$(1,1,0,0)$
H	$(6,7)$	4	19	23	19	23	$(0,0,0,0)$
I	$(7,8)$	2	23	25	23	25	$(0,0,0,0)$
K	$(3,4)$	0	3	3	10	10	$(7,7,0,0)$

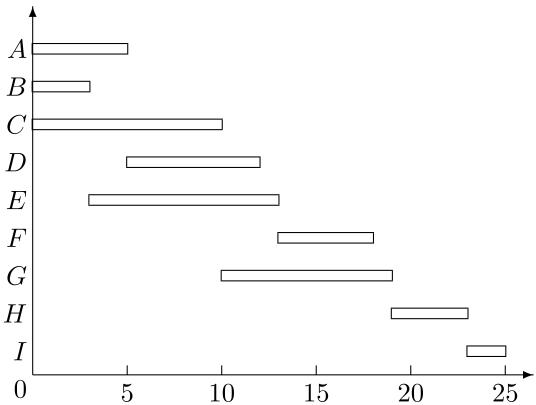


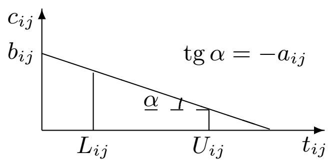
Рис. 7.3. Временная диаграмма проекта из примера 7.1

7.3. Распределение ресурсов в графиках проектов

Очень часто выполнение части, или даже всех работ проекта можно ускорить, если выделить дополнительные ресурсы. Допустим, что руководитель проекта может оценить продолжительность выполнения работы как функцию суммы денег, выделенных на ее выполнение.

Введем следующие обозначения:

- t_{ij} — продолжительность работы (i, j);
- L_{ij} — минимальная продолжительность работы (i, j);
- U_{ij} — максимальная (или «нормальная») продолжительность работы (i, j);
- $c_{ij}(t_{ij})$ — стоимость выполнения работы (i, j) за время t_{ij}.



Будем считать, что зависимость продолжительности работ от затрат линейная:

$$c_{ij}(t_{ij}) = b_{ij} - a_{ij}t_{ij},$$
$$\text{где } L_{ij} \le t_{ij} \le U_{ij}.$$

Для проекта, представленного сетевым графиком $G = (N, E)$, нужно построить график зависимости критического времени от суммы выделенных средств на выполнение проекта. Алгоритм решения данной задачи продемонстрируем на простом примере.

Пример 7.2. *Построить график зависимости критического времени от суммы выделенных средств для примера с исходными данными из табл. 7.3.*

1. Сетевой график с результатами рассчетов приведен на рис. 7.4. Первой точкой на графике будут точка с координатами $(C_1, T_1^{\text{кр}}) = (19, 10)$.

Строим орграф $G_1^{\text{кр}}$ (рис. 7.5), содержащий только критические дуги в текущем сетевом графике. Пропускные способности дуг определяются по правилу: если продолжительность выполнения работы (i, j) больше минимальной продолжительности, то пропускная способность дуги (i, j) равна a_{ij}; в противном случае пропускная способность дуги (i, j) равна ∞.

В построенной сети находим минимальный $(1, \bar{6})$-разрез $R_1 = \{(2, 4), (3, 4)\}$. Уменьшая на δ_1 продолжительность выполнения работ $(2, 4)$ и $(3, 4)$,

Таблица 7.3

Исходные давнные для примера 7.2

Работа	Продолжит. U_{ij}	Продолжит. L_{ij}	Непоср. предш.	Дуга	b_{ij}	a_{ij}
A	4	4	—	$(1,2)$	5	—
B	4	3	—	$(1,3)$	10	2
C	3	1	A	$(2,4)$	6	1
D	3	1	B	$(3,4)$	7	1
E	3	2	B	$(3,5)$	12	3
F	3	3	D, C	$(4,6)$	7	2
G	2	2	E	$(5,6)$	1	—

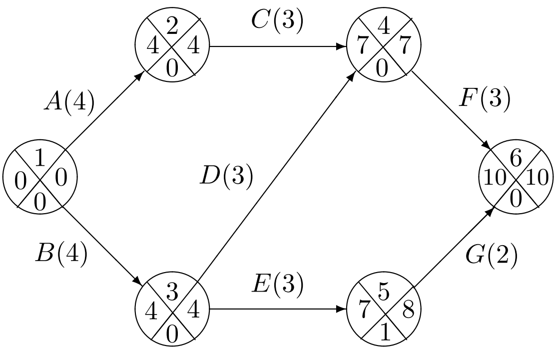


Рис. 7.4. Сетевой график на итерации 1

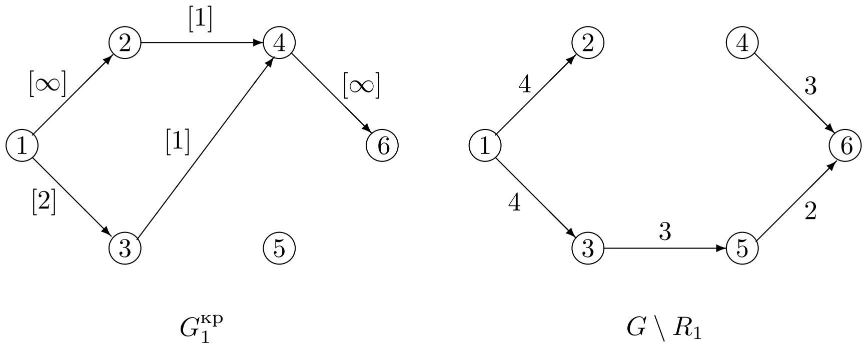


Рис. 7.5. Графы $G_1^{\text{кр}}$ и $G \setminus R_1$

мы уменьшим длину каждого критическог пути на δ_1. Но длины путей, которые не содержат ни одной из дуг $(2,4)$ или $(3,4)$, не изменятся. Поэтому, чтобы узнать, при каком δ в нашем сетевом графике появится новый критический путь, мы удаляем из графика обе дуги $(2,4)$ и $(3,4)$, и в полученном подграфе $G_1 \setminus R$ (рис. 7.5), длины дуг которого равны продолжительностям соответствующих работ, ищем путь максимальной длины из вершины 1 в вершину 6. Это путь $P_1 = (1,3,5,6)$ длины $L_1^{\text{кр}} = 9$. Поэтому путь P_1 станет критическим, если мы уменьшим продолжительность работ $(2,4)$ и $(3,4)$ на величину $\delta_1 = 9$.

Мы можем уменьшить продолжительность выполнения работы $(2,4)$ на $\delta_2 = 2$: с текущей продолжительности, равной 3, до минимальной, равной 1. Аналогично, мы можем уменьшить продолжительность выполнения работы $(3,4)$ на $\delta_3 = 2$: с текущей продолжительности, равной 3, до минимальной, равной 1. Определяем величину $\delta = \min\{\delta_1, \delta_2, \delta_3\} = \min\{1,2,2\} = 1$, на которую будем уменьшать продолжительность работ $(2,4)$ и $(3,4)$.

2. На данной и последующей итерациях действуем также, как и на итерации 1. Пересчитываем параметры сетевого графика для измененных продолжительностей работ t_{ij} (рис. 7.6). Получаем $C_2 = C_1 + 2 = 21$ и $T_2^{\text{кр}} = 9$.

Затем, в графе $G_2^{\text{кр}}$ (рис. 7.7) находим минимальный разрез $R_2 = \{(2,4),(1,3)\}$. Строим граф $G \setminus R_2$ (рис. 7.7) и вычисляем

$$L^2_{\text{кр}} = -\infty,\ \delta_1 = \infty,\ \delta_2 = \min\{3-2, 4-3\} = 1,$$
$$\delta = 1,\ t_{24} = 2-1 = 2,\ t_{13} = 4-1 = 3.$$

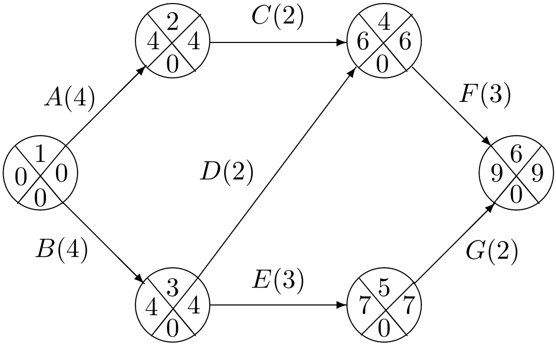


Рис. 7.6. Сетевой график на итерации 2

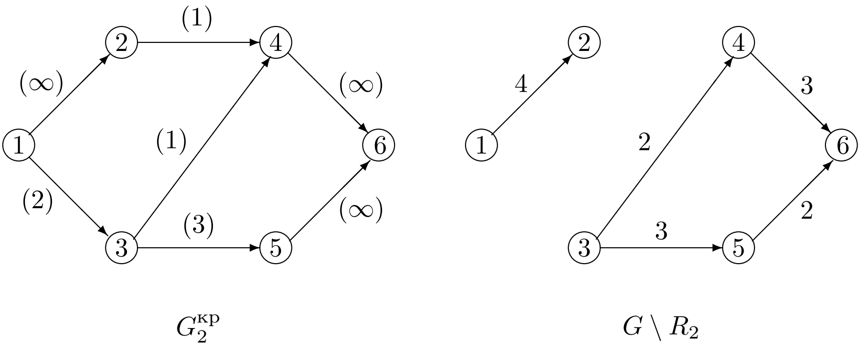


Рис. 7.7. Графы $G_2^{\text{кр}}$ и $G \setminus R_2$

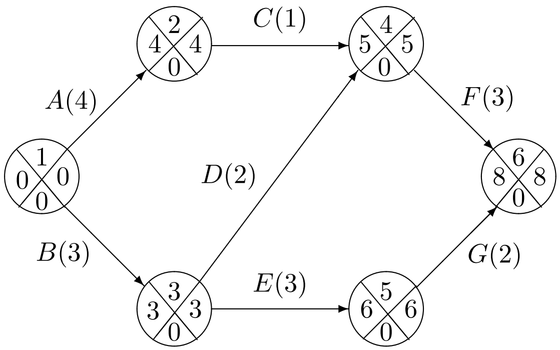


Рис. 7.8. Сетевой график на итерации 3

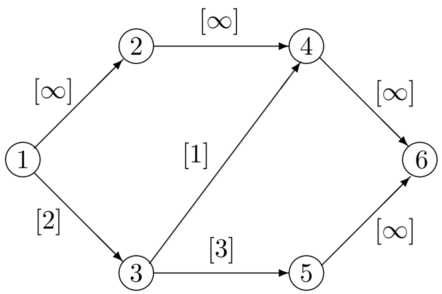


Рис. 7.9. Граф $G_3^{\text{кр}}$

Пересчитываем параметры сетевого графика для измененных продолжительностей работ t_{ij} (рис. 7.8). Получаем $C_3 = C_2 + 3 = 24$ и $T_3^{\text{кр}} = 8$.

Строим граф $G_3^{\text{кр}}$ (рис. 7.9) и находим минимальный разрез в нем $R_3 = \{(2,4),(3,4),(3,5)\}$, который имеет величину ∞. Работа алгаритма завершена. График зависимости критического времени от суммы дополнительных средств представлен на рис. 7.10.

□

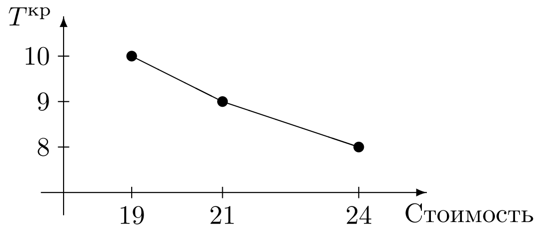


Рис. 7.10. График зависимости критического времени от стоимости реализации проекта

7.4. Упражнения

7.1. Постройте сетевой график и вычислите его параметры для проектов, описанных в следующих таблицах:

а)

Работа	Прод.	Непоср. предш.
A	10	-
B	6	-
C	7	-
D	5	B,C
E	6	A,B,C
F	8	A
G	4	F
H	10	E
I	5	H,D
J	7	F

б)

Работа	Прод.	Непоср. предш.
A	8	-
B	6	-
C	5	-
D	7	A,C
E	9	A,B,C
F	8	B
G	3	F
H	2	D,E,G
I	10	A,C
J	7	F

в)

Работа	A	B	C	D	E	F	G	H	I	J	K	L
Прод.	9	10	7	5	13	3	8	10	4	17	3	6
Непоср. предш.	-	-	-	C	A,B,C	A	C	G	E,F	E,F	I,H	G

7.2. Постройте кривую зависимости между затратами и продолжительностью проекта для проектов, исходные данные для которых приведены в следующих таблицах.

а)

Работа	Непоср. предш.	Продолжит.		b_{ij}	a_{ij}
		U_{ij}	L_{ij}		
A	—	5	3	20	3
B	—	7	4	53	7
C	—	6	4	56	8
D	C	3	3	10	0
E	C	5	3	25	4
F	B	12	8	60	4
G	A,C	8	5	45	5
H	E	6	6	5	0

б)

Работа	Непоср. предш.	Продолжит.		b_{ij}	a_{ij}
		U_{ij}	L_{ij}		
A	—	5	2	20	3
B	—	3	1	9	2
C	—	6	4	15	2
D	A	4	2	24	5
E	A	7	3	10	1
F	B,D	3	1	16	4
G	A,C	3	3	4	0

Глава 8
Задачи с неопределенными параметрами

Формулировки многих оптимизационных задач включают неопределенные параметры. Имеется несколько подходов к решению таких задач. В *стохастическом программировании* предполагается, что все неопределенные параметры являются случайными величинами с известными распределениями вероятностей. *Робастная оптимизация* используется тогда, когда требуется, чтобы решение было приемлемым для всех возможных значений неопределенных параметров. Последнее очень важно в тех ситуациях, когда небольшие изменения исходных данных задачи могут привести к тому, что ее решение меняется кардинальным образом.

Обычно решение оптимизационной задачи с неопределенными параметрами сводится к решению ее детерминированного эквивалента, который, как правило, является оптимизационной задачей гораздо большего размера, чем исходная задача. В этой главе мы будем изучать только такие модели стохастического программирования и робастной оптимизации, детерминированный вариант которых является задачей СЦП.

8.1. Двустадийные задачи стохастического программирования

Модели стохастического программирования могут включать два типа переменных: ожидаемые и адаптивные переменные. *Ожидаемые переменные* представляют те решения, которые нужно принять *здесь-и-сейчас*: они не зависят от будущей реализации случайных параметров. Решения, соответствующие *адаптивным переменным*, принимаются после того, как станут известны значения случайных параметров. Для примера, рассмотрим *двустадийную задачу стохастического программирования*, которая формулируется следующим образом:

$$\begin{array}{rrl} c^T x + & E(h(\omega)^T y(\omega)) & \to \max, \\ A(\omega)\, x + & G(\omega)\, y(\omega) & \le b(\omega), \\ x & & \in X, \\ & y(\omega) & \in \mathbb{R}^{n_y}_+. \end{array} \tag{8.1}$$

В этой формулировке решение для использования в текущем временном периоде представлено вектором $x \in X$ ожидаемых переменных, где $X \subseteq R^{n_x}$ некоторое множество (например, $X = \mathbb{R}^{n_x}_+$, $X = \mathbb{Z}^{n_x}_+$ или $X = P(A_0, b_0; S)$). Решение $x \in X$ нужно принять до того, как в следующем периоде реализуется элементарное событие ω из вероятностного пространства $(\Omega, \mathcal{A}, \mathbb{P})$. Решение $y(\omega)$ принимается в этом следующем периоде после наблюдения события ω. Поэтому вектор y адаптивных переменных есть функция от ω. Система $A(\omega)x + G(\omega)y(\omega) \le b(\omega)$ стохастических ограничений связывет ожидаемые и адаптивные переменные. Целевая функция в задаче (8.1) есть сумма двух членов: детерминированного $c^T x$, оценивающего качество решения x, и ожидаемого значения $E(h(\omega)^T y(\omega))$ случайной величины $h(\omega)^T y(\omega)$, оценивающей качество решения $y(\omega)$.

Задачу (8.1) можно переформулировать следующим образом:

$$\max\{f(x) :\ x \in X\}, \tag{8.2}$$

где $f(x) = E(f(x, \omega))$, а случайная величина $f(x, \omega)$ (см. упр. 8.2) определяется по правилу:

$$\begin{aligned} f(x, \omega) &\stackrel{\text{def}}{=} c^T x + \max h(\omega)^T y(\omega), \\ G(\omega) y(\omega) &\le b(\omega) - A(\omega) x, \\ y(\omega) &\in \mathbb{R}^{n_y}_+. \end{aligned} \tag{8.3}$$

Если выборочное пространство бесконечное, то вычисление $f(x)$ может быть очень сложной задачей. Один из подходов состоит в том, чтобы аппроксимировать бесконечное вероятностное пространство конечным пространством. Обсуждение того, как это делается, выходит за рамки данной книги. В дальнейшем мы будем предполагать, что $\Omega = \{\omega_1, \dots, \omega_K\}$ есть конечное множество с распределением вероятностей $p = (p_1, \dots, p_K)^T$, т. е. событие (сценарий) ω_k случается с вероятностью p_k. Для $k = 1, \dots, K$ введем обозначения: $h_k = h(\omega_k)$, $w_k = p_k h_k$,

$A_k = A(\omega_k)$, $G_k = G(\omega_k)$, $b_k = b(\omega_k)$, $y_k = y(\omega_k)$, $n_k = n_y$. *Детерминированный эквивалент* стохастической задачи (8.1) записывается следующим образом:

$$\begin{array}{rcll} c^T x + \sum_{k=1}^{K} w_k^T y_k & \to & \max, & \\ A_k x + \quad G_k y_k & \le & b_k, & k = 1, \dots, K, \\ x & \in & X, & \\ y_k & \in & \mathbb{R}_+^{n_k}, & k = 1, \dots, K. \end{array} \tag{8.4}$$

Решив задачу (8.4), мы получим решение x для использования в текущем временном периоде. Решение x должно быть адекватным всему, что может случиться в следующем периоде. Если бы мы знали, какой сценарий ω_k случится в следующем периоде, мы бы решили задачу

$$\begin{array}{rcl} c^T x + h_k^T y_k & \to & \max, \\ A_k x + G_k y_k & \le & b_k, \\ x & \in & X, \\ y_k & \in & \mathbb{R}_+^{n_k}, \end{array}$$

которая учитывает ограничения только для данного сценария. Но поскольку мы не знаем, какой из сценариев реализуется, то в задаче (8.4) мы требуем выполнения ограничений $A_k x + G_k y_k \le b_k$ для всех сценариев $k = 1, \dots, K$.

8.2. Минимизация рисков

Максимизация ожидаемой прибыли предполагает повторяемость процесса принятия решения достаточно большое количество раз при одинаковых условиях. Только тогда асимптотические утверждения, такие, как закон больших чисел, гарантируют сходимость в вероятностных терминах случайных величин к их ожидаемым значениям. В других ситуациях мы не можем не учитывать риск получения прибыли, которая существенно меньше ожидаемого значения. Определение подходящих мер риска является предметом активных исследований. Мы не собираемся исследовать проблему моделирования рисков во всей полноте, а лишь обсудим одно понятие риска, которое удобно для использования в СЦП моделях.

Здесь мы постараемся расширить двустадийную модель стохастического программирования (8.1), добавив к ней систему неравенств, ограничивающую риск принимаемого решения x. Понятие риска удобнее вводить в терминах функции потерь $g(x, \omega)$, которая зависит от решения x и является случайной величиной, определенной на некотором вероятностном пространстве $(\Omega, \mathcal{A}, \mathbb{P})$ $(\omega \in \Omega)$.

Исторически первое и, пожалуй, самое известное понятие риска ввел Нобелевский лауреат в области экономики 1990 г. Х. Марковиц. Он определил *риск* как вариацию дохода (или потерь):

$$\mathrm{var}(g(x, \omega)) = E((g(x, \omega) - E(g(x, \omega)))^2).$$

Концептуально такое понятие риска имеет несколько недостатков. Самый важный из них в том, что данная мера риска симметрична: она одинаково штрафует за получение как меньших, так и больших потерь, чем ожидаемое значение. С точки зрения использования в СЦП недостатком является и то, что ограничение таких рисков означает введение в модель задачи квадратичных (нелинейных) ограничений.

Еще одна не менее известная мера риска под названием VaR (Value-at-Risk) была разработана финансовыми инженерами компании J. P. Morgan. Пусть

$$G(x, \eta) \stackrel{\mathrm{def}}{=} \mathbb{P}\{\omega \in \Omega : \; g(x, \omega) \leq \eta\}$$

есть функция распределения случайной величины $g(x, \omega)$. Для заданной вероятности $0 < \alpha < 1$ мера VaR_α определяется по формуле

$$\mathrm{VaR}_\alpha(x) \stackrel{\mathrm{def}}{=} \min\{\eta : \; G(x, \eta) \geq \alpha\}.$$

Для дискретного вероятностного пространства $\Omega = \{\omega_1, \ldots, \omega_K\}$, чтобы вычислить $\mathrm{VaR}_\alpha(x)$, нужно

1) записать значения $g_k(x) \stackrel{\mathrm{def}}{=} g(x, \omega_k)$ в возрастающем порядке:
$$g_{\pi(1)}(x) \leq g_{\pi(2)}(x) \leq \cdots \leq g_{\pi(K)}(x);$$

2) найти минимальный индекс j такой, что $\sum_{i=1}^{j} p_{\pi(i)} \geq \alpha$, и положить $\mathrm{VaR}_\alpha(x) = g_{\pi(j)}(x)$.

Для примера, рассмотрим дискретное вероятностное пространство с

$$\Omega = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\},$$

когда первые 5 событий $(0, 1, 2, 3, 4)$ происходят с вероятностью $1/20$, а остальные события $(5, 6, 7, 8, 9)$ — с вероятностью $3/20$. Пусть $\alpha = 0.9$ и $g(x, \omega) = x - \omega$ для $x \in \mathbb{Z}_+$. Поскольку $\omega_k = k - 1$, то для $x = 5$ имеем $g_k(5) = g(5, k - 1) = 6 - k$, $k = 1, \ldots, 10$. Сортируем значения $g_k(5)$:

i	1	2	3	4	5	6	7	8	9	10
$\pi(i)$	10	9	8	7	6	5	4	3	2	1
$g_{\pi(i)}(5)$	-4	-3	-2	-1	0	1	2	3	4	5
$p_{\pi(i)}$	$\frac{3}{20}$	$\frac{3}{20}$	$\frac{3}{20}$	$\frac{3}{20}$	$\frac{3}{20}$	$\frac{1}{20}$	$\frac{1}{20}$	$\frac{1}{20}$	$\frac{1}{20}$	$\frac{1}{20}$

Поскольку

$$\sum_{i=1}^{8} p_{\pi(i)} = 5\,\frac{3}{20} + 3\,\frac{1}{20} = 0.9,$$

то $j = 8$ и $\mathrm{VaR}_{0.9}(5) = g_{\pi(8)}(5) = g_3(5) = 3$.

Мера риска VaR широко используется в финансовой индустрии, и ее вычисление является одним из стандартных атрибутов большинства программ финансового анализа. Несмотря на свою популярность, мера VaR также не без недостатков. Одним из таких недостатков является то, что эта мера никак не оценивает величину потерь, превосходящих $\mathrm{VaR}_\alpha(x)$. Другим недостатком меры VaR является то, что функция $\mathrm{VaR}_\alpha(x)$ не является супераддитивной ($\mathrm{VaR}_\alpha(x+y) \le \mathrm{VaR}_\alpha(x) + \mathrm{VaR}_\alpha(y)$). В финансовой терминологии супераддитивность выражает тот факт, что диверсификация инвестиций снижает риск. Еще один недостаток меры VaR заключается в том, что функцию $\mathrm{VaR}_\alpha(x)$ трудно использовать в оптимизации, поскольку она не является выпуклой. Эти и другие недостатки меры VaR мотивировали появление ряда ее модификаций.

Содержательно, мера $\mathrm{CVaR}_\alpha(x)$ (Conditional-Value-at-Risk) определяется как ожидаемые (средние) потери, при условии, что эти потери не меньше $\mathrm{VaR}_\alpha(x)$. Формально, $\mathrm{CVaR}_\alpha(x)$ определяется как матожидание случайной величины $g(x,\omega)$ с так называемым α-хвостовым распределением

$$G_\alpha(x,\eta) \stackrel{\mathrm{def}}{=} \begin{cases} 0, & \text{если } \eta < \mathrm{VaR}_\alpha(x), \\ \dfrac{G(x,\eta)-\alpha}{1-\alpha}, & \text{если } \eta \ge \mathrm{VaR}_\alpha(x). \end{cases}$$

По определению $\mathrm{CVaR}_\alpha(x) = \int_{-\infty}^{\infty} \eta\, dG_\alpha(x,\eta)$. Данное определение трудно использовать в вычислениях. Следующее утверждение снимает это затруднение.

Теорема 8.1. *Справедливы равенства*

$$\mathrm{CVaR}_\alpha(x) = \min_{\eta\in\mathbb{R}} g_\alpha(x,\eta) = g_\alpha(x, \mathrm{VaR}_\alpha(x)),$$

где

$$g_\alpha(x,\eta) \stackrel{\text{def}}{=} \eta + \frac{1}{1-\alpha}\int_\Omega \max\{g(x,\omega)-\eta, 0\}\mathbb{P}(d\omega).$$

Как и ранее, здесь мы также ограничимся рассмотрением сценарного подхода. Поэтому снова предположим, что $\Omega = \{\omega_1,\dots,\omega_K\}$ есть конечное множество с распределением вероятностей $p = (p_1,\dots,p_K)^T$. В этом случае

$$g_\alpha(x,\eta) = \eta + \frac{1}{1-\alpha}\sum_{k=1}^K p_k \max\{g_k(x)-\eta, 0\}, \tag{8.5}$$

где $g_k(x) \stackrel{\text{def}}{=} g(x,\omega_k)$.

Продолжая пример, в котором мы вычислили величину $\mathrm{VaR}_{0.9}(5) = 3$, вычислим

$$\begin{aligned}\mathrm{CVaR}_{0.9}(5) &= g_{0.9}(5, \mathrm{VaR}_{0.9}(5)) = g_{0.9}(5,3)\\ &= 3 + \frac{1}{1-0.9}\cdot\frac{1}{20}\cdot(2+1) = 3 + \frac{3}{2} = 4.5.\end{aligned}$$

8.2.1. Расширенная двустадийная модель

Снова рассмотрим двустадийную задачу (8.1). Но теперь мы хотим максимизировать ожидаемую прибыль при ограниченном риске, требуя выполнения неравенства $\mathrm{CVaR}_\alpha(x) \le r$, где r есть максимально допустимый уровень риска. Вводя переменные z_k для представления $\max\{g_k(x)-\eta, 0\}$ в формуле (8.5), мы можем расширить детерминированный вариант (8.4) задачи (8.1) следующим образом:

$$\begin{aligned}
&c^Tx + \sum_{k=1}^K w_k^T y_k \to \max,\\
&A_kx + G_ky_k \le b_k, \qquad k=1,\dots,K,\\
&\eta + \frac{1}{1-\alpha}\sum_{k=1}^K p_k z_k \le r,\\
&g_k(x) - \eta - z_k \le 0, \qquad k=1,\dots,K,\\
&\eta\in\mathbb{R},\ x\in X,\\
&z_k \ge 0,\ y_k \in \mathbb{R}_+^{n_k}, \qquad k=1,\dots,K.
\end{aligned} \tag{8.6}$$

Задача (8.6) будет задачей СЦП, если функция $g_k(x)$ линейная и $X = P(A,b;S)$. Особый интерес представляет также случай, когда $g(x,\omega) =$

$-f(x,\omega)$ (заметим, что $f(x,\omega)$ есть нелинейная функция). Тогда задачу (8.6) можно переписать следующим образом:

$$
\begin{aligned}
c^T x + \sum_{k=1}^{K} w_k^T y_k &\to \max, \\
A_k x + G_k y_k &\le b_k, \qquad k = 1,\dots,K, \\
\eta + \frac{1}{1-\alpha}\sum_{k=1}^{K} p_k z_k &\le r, \\
c^T x + h_k^T y_k + \eta + z_k &\ge 0, \qquad k = 1,\dots,K, \\
\eta \in \mathbb{R},\ x &\in X, \\
z_k \ge 0,\ y_k &\in \mathbb{R}_+^{n_k}, \qquad k = 1,\dots,K.
\end{aligned}
\tag{8.7}
$$

Убедиться в эквивалентности задач (8.6) и (8.7) несложно. Достаточно заметить, что по определению

$$g_k(x) = -c^T x - \max\{h_k^T y_k : \ G_k y_k \le b_k - A_k x\}$$

и поскольку

$$c^T x + \sum_{k=1}^{K} w_k^T y_k = \sum_{k=1}^{K} p_k (c^T x + h_k^T y_k),$$

то в оптимальном решении $(x^*; y_1^*, \dots, y_K^*; \eta^*; z_1^*, \dots, z_K^*)$ задачи (8.7) вектор y_k^* является оптимальным решением задачи

$$\max\{h_k^T y_k : \ G_k y_k \le b_k - A_k x\}$$

и, следовательно, $g_k(x^*) = -c^T x - h_k^T y_k^*$.

8.2.2. Кредитный риск

Кредитный риск — это риск, обусловленный тем, что партнеры в полной мере не выполняют своих обязательств. Потери появляются из-за отказа партнеров выполнять свои обязательства или из-за уменьшения рыночной цены активов, обусловленного падением кредитных рейтингов. Например, портфель облигаций из развивающихся рынков (Бразилия, Россия, Малайзия и др.) может с большой вероятностью приносить доход, но одновременно имеется небольшая вероятность больших потерь. Для таких инвестиций функции распределения возвратов (будущих доходов) несимметричны и, следовательно, симметричные меры риска здесь не совсем уместны. А вот мера CVaR, по сути, и была изобретена для оценки рисков в подобных ситуациях.

Рассмотрим задачу оптимизации портфеля из n потенциальных инвестиций (таких, как акции). Нам нужно определить долю средств x_j для вложения в инвестицию j. Тогда портфель характеризуется вектором $x = (x_1, \ldots, x_n)^T$. Множество X допустимых решений (портфелей) описывается системой

$$\sum_{j=1}^{n} x_j = 1,$$
$$l_j \le x_j \le u_j, \quad j = 1, \ldots, n.$$

Рассматривается K возможных сценариев по завершению планового горизонта. Пусть p_k есть вероятность появления сценария k, а μ^k есть вектор возвратов для этого сценария, где μ_j^k есть возврат (на один вложенный доллар) инвестиции j. Тогда $\mu = \sum_{k=1}^{K} p_k \mu^k$ есть вектор ожидаемых возвратов.

Потери портфеля x, если случится сценарий k, определяются по формуле: $g_k(x) = (q - \mu^k)^T x$, где q есть вектор возвратов, при условии, что кредитный рейтинг каждой инвестиции не изменится. Мы определяем риск портфеля x равным $\mathrm{CVaR}_\alpha(x)$ и ограничиваем этот риск заданной величиной r. Заметим, что при такой постановке «уровень безопасности» нашего решения (портфеля) x определяется выбором двух параметров α и r.

При сделанных выше предположениях задача максимизации ожидаемого возврата портфеля при ограниченном риске записывается следующим образом:

$$\mu^T x \to \max, \tag{8.8a}$$
$$\eta + \frac{1}{1-\alpha} \sum_{k=1}^{K} p_k z_k \le r, \tag{8.8b}$$
$$(q - \mu^k)^T x - \eta - z_k \le 0, \quad k = 1, \ldots, K, \tag{8.8c}$$
$$\sum_{j=1}^{n} x_j = 1, \tag{8.8d}$$
$$l_j \le x_j \le u_j, \quad j = 1, \ldots, n, \tag{8.8e}$$
$$z_k \ge 0, \quad k = 1, \ldots, K, \tag{8.8f}$$
$$\eta \in \mathbb{R}. \tag{8.8g}$$

Заметим, что (8.8) — это задача ЛП. Однако она может превратиться в задачу СЦП после учета ряда стандартных для портфельной оптимизации дополнительных логических условий. Одним из таких условий является требование диверсификации инвестиций. Предположим, что множество $N = \{1, \dots, n\}$ разбито на подмножества $N_1, \dots, N_m$, скажем, по отраслевому или территориальному принципу. Требуется, чтобы из группы N_i в портфеле присутствовало не более n_i различных инвестиций, а также чтобы в портфеле были инвестиции не менее чем из s групп.

Введем две группы бинарных переменных:

- $y_j = 1$, если инвестиция j присутствует в портфеле, и $y_j = 0$ в противном случае ($j = 1, \dots, n$);
- $\delta_i = 1$, если хотя бы одна инвестиция группы i присутствует в портфеле, и $\delta_i = 0$ в противном случае ($i = 1, \dots, m$).

Для учета перечисленных требований из задачи (8.8) нужно удалить неравенства (8.8e) и добавить следующие ограничения:

$$
\begin{aligned}
l_j y_j \le x_j \le u_j y_j, &\quad j = 1, \dots, n,\\
\sum_{j \in N_i} y_i \le n_i, &\quad i = 1, \dots, m,\\
y_j \le \delta_i, &\quad j \in N_i,\ i = 1, \dots, m,\\
\sum_{i=1}^{m} \delta_i \ge s, &\\
y_j \in \{0, 1\}, &\quad j = 1, \dots, n,\\
\delta_i \in \{0, 1\}, &\quad i = 1, \dots, m.
\end{aligned}
$$

8.2.3. Портфель из трех активов

Предположим, что мы собираемся сформировать портфель из трех активов, характеристики которых описаны в табл. 8.1.

В конце планового горизонта (через один год) рейтинг фирмы-эмитента может измениться. Для каждого из возможных значений будущего рейтинга по простым формулам мы можем рассчитать возврат (на один вложенный доллар) каждого из активов. Значения этих будущих возвратов представлены в табл. 8.2. Возвраты активов при дефолте являются

Таблица 8.1

Характеристики активов

Эмитент	Рейтинг	Срок обращения (лет)	Годовая проц. ставка
Фирма 1	BBB	5	6 %
Фирма 2	A	5	5 %
Фирма 3	CCC	2	10 %

Таблица 8.2

Возвраты активов (на \$1)

Будущий рейтинг	Фирма 1	Фирма 2	Фирма 3
AAA	1.0937	1.0659	1.162
AA	1.0919	1.0649	1.161
A	1.0866	1.0630	1.161
BBB	1.0755	1.0564	1.157
BB	1.0202	1.0315	1.142
B	0.9810	1.0139	1.137
CCC	0.8364	0.8871	1.056

Таблица 8.3

Возвраты активов при дефолте (на \$1)

Восстановление при дефолте	Фирма 1	Фирма 2	Фирма 3
Среднее значение	0.5113	0.5113	0.530
Стандартное отклонение	0.25	0.25	0.33

Таблица 8.4

Вероятности переходов

Рейтинг	Фирма 1	Фирма 2	Фирма 3
AAA	0.0002	0.0009	0.0022
AA	0.0033	0.0227	0.0000
A	0.0595	$\boxed{0.9105}$	0.0022
BBB	$\boxed{0.8693}$	0.0552	0.0130
BB	0.0530	0.0074	0.0238
B	0.0117	0.0026	0.1124
CCC	0.0012	0.0001	$\boxed{0.6486}$
Дефолт	0.0018	0.0006	0.1979

Таблица 8.5

Корреляционная матрица

	Фирма 1	Фирма 2	Фирма 3
Фирма 1	1.0	0.3	0.1
Фирма 2	0.3	1.0	0.2
Фирма 3	0.1	0.2	1.0

случайными величинами с бета распределением, параметры которого представлены в табл. 8.3.

Будущий (через год) рейтинг каждой из фирм есть случайная величина с распределением вероятностей, представленным в соответствующем столбце табл. 8.4. Заметим, что эти вероятности переходов всегда можно узнать на сайте соответствующего рейтингового агентства. Взаимозависимости случайных возвратов наших трех активов заданы корреляционной матрицей из табл. 8.5.

В данном приложении каждый будущий сценарий описывается указанием рейтингов фирм. Например, сценарий (BB, BBB, CCC) означает, что в будущем фирма 1 будет иметь рейтинг BB, фирма 2 — BBB, фирма 3 — CCC.

Для генерации сценариев нужно по вероятностям переходов вычис-

Таблица 8.6

Пороговые значения

Порог	Фирма 1	Фирма 2	Фирма 3
Z_{AA}	3.54	3.12	2.86
Z_{A}	2.78	1.98	2.86
Z_{BBB}	1.53	−1.51	2.63
Z_{BB}	−1.49	−2.30	2.11
Z_{B}	−2.18	−2.72	1.74
Z_{CC}	−2.75	−3.19	1.02
Z_{def}	−2.91	−3.24	−0.85

лить пороговые значения по следующим формулам:

$$\begin{aligned}
p_{\mathrm{AAA}} &\stackrel{\mathrm{def}}{=} 1-\Phi(Z_{\mathrm{AA}}), & p_{\mathrm{AA}} &\stackrel{\mathrm{def}}{=} \Phi(Z_{\mathrm{AA}})-\Phi(Z_{\mathrm{A}}),\\
p_{\mathrm{A}} &\stackrel{\mathrm{def}}{=} \Phi(Z_{\mathrm{A}})-\Phi(Z_{\mathrm{BBB}}), & p_{\mathrm{BBB}} &\stackrel{\mathrm{def}}{=} \Phi(Z_{\mathrm{BBB}})-\Phi(Z_{\mathrm{BB}}),\\
p_{\mathrm{BB}} &\stackrel{\mathrm{def}}{=} \Phi(Z_{\mathrm{BB}})-\Phi(Z_{\mathrm{B}}), & p_{\mathrm{B}} &\stackrel{\mathrm{def}}{=} \Phi(Z_{\mathrm{B}})-\Phi(Z_{\mathrm{CCC}}),\\
p_{\mathrm{CCC}} &\stackrel{\mathrm{def}}{=} \Phi(Z_{\mathrm{CCC}})-\Phi(Z_{\mathrm{def}}), & p_{\mathrm{def}} &\stackrel{\mathrm{def}}{=} \Phi(Z_{\mathrm{def}}),
\end{aligned}$$

где

$$\Phi(z) \stackrel{\mathrm{def}}{=} \frac{1}{2\sqrt{\pi}} \int_{-\infty}^{z} \mathbf{e}^{-r^2/2} \mathrm{d}r$$

есть функция распределения нормальной случайной величины с матожиданием 0 и стандартным отклонением 1.

Для рассматриваемых фирм-эмитентов пороговые значения представлены в табл. 8.6. Геометрическая интерпретация пороговых значений приведена на рис. 8.1. Здесь горизонтальные линии, проведенные через точки

$$Z_{\mathrm{AA}}, Z_{\mathrm{A}}, Z_{\mathrm{BBB}}, Z_{\mathrm{BB}}, Z_{\mathrm{B}}, Z_{\mathrm{CCC}}, Z_{\mathrm{def}},$$

разбивают фигуру под «колоколом» нормального распределения на части, площадь которых равна

$$p_{\mathrm{AAA}}, p_{\mathrm{AA}}, p_{\mathrm{A}}, p_{\mathrm{BBB}}, p_{\mathrm{BB}}, p_{\mathrm{B}}, p_{\mathrm{CCC}} \text{ и } p_{\mathrm{def}}.$$

Чтобы сгенерировать сценарий, мы сначала генерируем нормально распределенный случайный вектор $\xi = (\xi_1, \xi_2, \xi_3)$ с нулевым матожиданием и ковариционной матрицей Σ, которая совпадает с корреляционной матрицей для наших трех активов. Это можно сделать, сначала

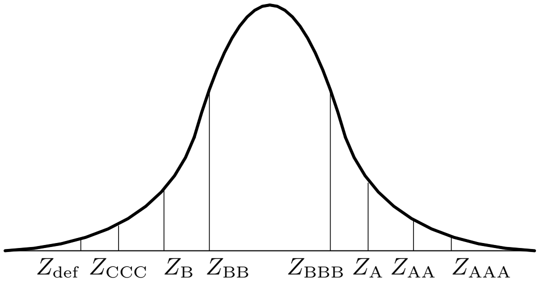


Рис. 8.1. Пороговые значения рейтингов

сгенерировав вектор из трех независимых нормальных случайных величин с матожиданием 0 и стандартным отклонением 1, а затем результат умножить на матрицу B такую, что $\Sigma = B^T B$. Такую матрицу B можно получить, вычислив, например, разложение Холесского для матрицы Σ.

Далее, по случайному вектору ξ мы записываем сценарий, определяя рейтинг фирмы $i = 1, 2, 3$ по минимальному пороговому значению, которое не меньше ξ_i; если, скажем, это Z_{BB}, то в данном сценарии фирма i будет иметь рейтинг BB. Для примера, рассмотрим строку 5 в табл. 8.7, в которой представлены 10 случайно сгенерированных сценариев. Здесь $\xi = (0.4690, -0.5639, 0.2832)$. Смотрим в табл. 8.6 пороговые значения для фирмы 1 и определяем, что $Z_{\mathrm{BB}} < \xi_1 \le Z_{\mathrm{BBB}}$. Значит, в пятом сценарии фирма 1 должна иметь рейтинг BBB. Аналогично, по столбцам 2 и 3 табл. 8.6 определяем, что $Z_{\mathrm{BBB}} < \xi_2 \le Z_{\mathrm{A}}$ и $Z_{\mathrm{def}} < \xi_3 \le Z_{\mathrm{CCC}}$. Следовательно, фирма 2 имеет рейтинг A, а фирма 3 — CCC.

Для каждого из сценариев нам осталось определить вектор возвратов. Для всех состояний, кроме дефолта, это делается совсем просто. Для примера, рассмотрим сценарий 2. По рейтингам фирм BB, BBB и CCC в табл. 8.2 находим возвраты 1.0202, 1.0564 и 1.056. При дефолте возврат фирмы генерируется в соответствии с заданным бета распределением (см. табл. 8.3). Значения возвратов μ^k для всех сценариев $k = 1, \dots, 10$ представлены в табл. 8.8.

Если в будущем рейтинги фирм не изменятся, то мы будем иметь сценарий (BBB, A, CCC) с вектором возвратов

$$q = (1.0755, 1.0630, 1.056).$$

Разности $q - \mu^k$ представлены в последних трех столбцах табл. 8.8.

Поскольку все сценарии равновероятны, то все $p_k = 0.1$, и мы можем

Таблица 8.7

Сценарии

Сценарий	Случайный вектор			Будущий рейтинг		
	ξ_1	ξ_2	ξ_3	Фирма 1	Фирма 2	Фирма 3
1	-0.7769	-0.8750	-0.6874	BBB	A	CCC
2	-2.1060	-2.0646	0.2996	BB	BBB	CCC
3	-0.9276	0.0606	2.7068	BBB	A	A
4	0.6454	-0.1532	-1.1510	BBB	A	дефолт
5	0.4690	-0.5639	0.2832	BBB	A	CCC
6	-0.1252	-0.5570	-1.9479	BBB	A	дефолт
7	0.6994	1.5191	-1.6503	BBB	A	дефолт
8	1.1778	-0.6342	-1.7759	BBB	A	дефолт
9	1.8480	2.1202	1.1631	A	AA	B
10	0.0249	-0.4642	0.3533	BBB	A	CCC

Таблица 8.8

Возвраты активов в сценариях

Сценарий	μ^k			$a^k = q - \mu^k$		
	μ_1^k	μ_2^k	μ_3^k	a_1^k	a_2^k	a_3^k
1	1.0755	1.0630	1.056	0	0	0
2	1.0202	1.0564	1.056	0.0553	0.0026	0
3	1.0755	1.0630	1.161	0	0	-0.105
4	1.0755	1.0630	0.657	0	0	0.399
5	1.0755	1.0630	1.056	0	0	0
6	1.0755	1.0630	0.754	0	0	0.302
7	1.0755	1.0630	0.269	0	0	0.787
8	1.0755	1.0630	0.151	0	0	0.905
9	1.0866	1.0649	1.137	-0.0111	-0.0019	-0.081
10	1.0755	1.0630	1.056	0	0	0

вычислить вектор μ средних возвратов, суммируя все десять векторов μ^k и деля результат на 10:

$$\mu = (1.0700, 1.0625, 0.8353).$$

Предположим, что $l_1 = l_2 = l_3 = 0.1$ и $u_1 = u_2 = u_3 = 0.5$. Теперь у нас есть все необходимые данные, и мы можем записать экземпляр задачи (8.8) следующим образом:

$$\begin{aligned}
1.07x_1 + 1.0625x_2 + 0.8353 \to \max,\\
\eta + \frac{1}{10(1-\alpha)}(z_1 + z_2 + z_3 + z_4 + z_5 + z_6 + z_7 + z_8 + z_9 + z_{10}) \le r,\\
-\eta - z_1 \le 0,\\
0.0553x_1 + 0.0026x_2 - \eta - z_2 \le 0,\\
-0.105x_3 - \eta - z_3 \le 0,\\
0.399x_3 - \eta - z_4 \le 0,\\
-\eta - z_5 \le 0,\\
0.302x_3 - \eta - z_6 \le 0,\\
0.787x_3 - \eta - z_7 \le 0,\\
0.905x_3 - \eta - z_8 \le 0,\\
-0.0111x_1 - 0.0019x_2 - 0.081x_3 - \eta - z_9 \le 0,\\
-\eta - z_{10} \le 0,\\
x_1 + x_2 + x_3 = 1,\\
0.1 \le x_1 \le 0.5,\ 0.1 \le x_2 \le 0.5,\ 0.1 \le x_3 \le 0.5,\\
z_1, z_2, z_3, z_4, z_5, z_6, z_7, z_8, z_9, z_{10} \ge 0.
\end{aligned}$$

Нам не стоит пытаться решить данную задачу ЛП, поскольку ее решение, скорее всего, ничего нам не скажет об оптимальном портфеле. Дело в том, что сценарии из табл. 8.7 недостаточно точно отражают вероятности переходов (например, дефолт фирмы 3 случается 4 раза из 10, т. е. с вероятностью 0.4, что ровно в два раза больше вероятности дефолта для этой фирмы), то 10 сценариев — это слишком мало даже для портфеля из трех активов. Чтобы обеспечить приемлемую точность расчетов в реальных примерах с десятками потенциальных инвестиций, приходится генерировать десятки тысяч сценариев.

8.3. Мультистадийные задачи стохастического программирования

Хотя мультистадийные модели стохастического программирования исследуются уже несколько десятилетий, они не использовались на практике для решения задач реалистичных размеров. Лишь только недавно с появлением достаточно мощных компьютеров модели стохастического программирования стали использоваться на практике, а само стохастическое программирование начало развиваться стремительными темпами.

Мультистадийные задачи стохастического программирования применяются, когда плановый горизонт включает более одного периода (стадии). Пусть T обозначает число периодов, а $\omega^t \in \Omega^t$ — событие, которое должно произойти в период $t = 1, \ldots, T$. В начале планового горизонта (на стадии 0) ожидаемое решение x должно приниматься в момент, когда событие ω^1 еще не произошло. Решение $y(\omega^1, \ldots, \omega^t)$ принимается на стадии t, когда события $\omega^1, \ldots, \omega^{t-1}$ уже произошли, а событие ω^t еще не реализовалось. Решение $y(\omega^1, \ldots, \omega^t)$ зависит от решения $y(\omega^1, \ldots, \omega^{t-1})$, принятого на предыдущей стадии. Мультистадийная модель записывается следующим образом:

$$
\begin{aligned}
c^T x + \sum_{t=1}^{T} h(\omega^1, \ldots, \omega^t)^T y(\omega^1, \ldots, \omega^t) &\to \max, \\
A(\omega^1)x + G(\omega^1)y(\omega^1) &\le b(\omega^1), \\
A(\omega^1, \omega^2)y(\omega^1) + G(\omega^1, \omega^2)y(\omega^1, \omega^2) &\le b(\omega^1, \omega^2), \\
A(\omega^1, \omega^2, \omega^3)y(\omega^1, \omega^2) + G(\omega^1, \omega^2, \omega^3)y(\omega^1, \omega^2, \omega^3) &\le b(\omega^1, \omega^2, \omega^3), \\
&\ddots \vdots \ddots \\
A(\omega^1, \ldots, \omega^T)y(\omega^1, \ldots, \omega^{T-1}) + G(\omega^1, \ldots, \omega^T)y(\omega^1, \ldots, \omega^T) &\le b(\omega^1, \ldots, \omega^T), \\
x &\in X, \\
y(\omega^1, \ldots, \omega^t) &\in \mathbb{R}^{n_y}, \quad t = \overline{1, T}.
\end{aligned}
\tag{8.9}
$$

Предположим, что для всех $t = 1, \ldots, T$ выборочное пространство Ω^t конечно, а последовательность событий

$$(\omega_1, \ldots, \omega_t) \in \Omega^1 \times \cdots \times \Omega^t$$

происходит с вероятностью $p(\omega_1, \ldots, \omega_t)$. Очевидно, должно выполняться равенство

$$\sum_{(\omega_1,\dots,\omega_t)\in\Omega^1\times\dots\times\Omega^t} p(\omega_1,\dots,\omega_t) = 1.$$

Поскольку некоторые из этих вероятностей могут быть нулевыми, чтобы сформулировать детерминированный эквивалент стохастической задачи (8.9), удобно ввести понятие *дерева сценариев.*

Узлы дерева сценариев занумерованы числами $0, 1, \dots, n$. Узел 0 — корень дерева. Узлы, находящиеся на расстоянии t от корня, принадлежат стадии t. Обозначим через $t(i)$ номер стадии узла i. Ориентируем ребра в направлении от корня к листьям, а ориентированные ребра назовем дугами. Заметим, что в любой узел, за исключением корня, входит только одна дуга (i, j), при этом узел i называют предком узла j и обозначают $parent(j)$. Каждой дуге (i, j) дерева приписано некоторое событие $\omega_{i,j}$ из $\Omega^{t(i)}$. Исходные данные задачи распределяются по узлам дерева следующим образом. Узлу 0 приписываются множество X и вектор $c_0 = c$. Каждому из остальных узлов j $(1 \le j \le n)$ приписываются следующие параметры:

$$\begin{aligned}
p_j &\stackrel{\text{def}}{=} p(\omega(0, i_1), \omega(i_1, i_2), \dots, \omega(i_{t(j)-1}, j)),\\
c_j &\stackrel{\text{def}}{=} h(\omega(0, i_1), \omega(i_1, i_2), \dots, \omega(i_{t(j)-1}, j)) \cdot p_j,\\
b_j &\stackrel{\text{def}}{=} b(\omega(0, i_1), \omega(i_1, i_2), \dots, \omega(i_{t(j)-1}, j)),\\
A_j &\stackrel{\text{def}}{=} A(\omega(0, i_1), \omega(i_1, i_2), \dots, \omega(i_{t(j)-1}, j)),\\
G_j &\stackrel{\text{def}}{=} G(\omega(0, i_1), \omega(i_1, i_2), \dots, \omega(i_{t(j)-1}, j)),
\end{aligned}$$

где последовательность $(0, i_1, \dots, i_{t(j)-1}, j)$ задает единственный путь в дереве, ведущий из корня 0 в узел j. Заметим, что по определению чисел p_j должны выполняться равенства:

$$\begin{aligned}
\sum_{j:\, t(j)=k} p_j &= 1,\\
\sum_{j:\, parent(j)=i} p_j &= p_i \quad \text{для всех } i \ge 0 \text{ таких, что } t(i) < T.
\end{aligned}$$

Обозначив через x_j вектор решений, принимаемых в узле j $(x_0 = x)$, мы записываем детерминированный эквивалент стохастической задачи

(8.9) следующим образом:

$$\begin{gathered}\sum_{j=0}^{n} c_j^T x_j \to \max, \\ A_j x_{parent(j)} + G_j x_j \le b_j, \quad j = 1, \ldots, n, \\ x_0 \in X, \\ x_j \ge 0, \quad j = 1, \ldots, n.\end{gathered} \tag{8.10}$$

Решив задачу (8.10), мы, в частности, найдем решение $x = x_0$, которое нужно принять в начале планового горизонта. Решения x_j в остальных узлах дерева сценариев на практике не реализуются. Когда закончится первый период, сегодняшнее будущее станет настоящим и мы уже будем знать, какой из сценарием первого уровня реализовался (будем знать значение события ω^1). Чтобы принять решение для реализовавшегося сценария, нужно построить новое дерево сценариев и решить новую задачу (8.10).

В следующих двух параграфах мы рассмотрим два конкретных примера мультистадийной задачи стохастического программирования.

8.3.1. Синтетические опционы

При формировании инвестиционного портфеля одной из важнейших целей является предотвращение падения доходности портфеля ниже критического уровня. Это можно сделать, включив в портфель производные финансовые активы, такие, как опционы. В ситуациях, когда производные активы недоступны, мы можем добиться нужного результата, формируя портфель на основе стратегии «синтетического опциона».

Исходные данные следующие:

- n — число рискованных активов;
- T — число периодов в плановом горизонте, период t начинается в момент времени $t-1$ и заканчивается в момент t;
- z_0 — сумма наличности в начале планового горизонта;
- x_{i0} — сумма, инвестированная в актив i в начале планового горизонта;
- R — процент на капитал ($1 +$ норма процента) в пересчете на один период;
- $r_{it} = r_i(\omega^1, \ldots, \omega^t)$ — случайный возврат (на один вложенный рубль) актива i в период t;

- ρ_{it} — стоимость транзакции при покупке и продаже актива i в период t; считаем, что все транзакции производятся в самом начале каждого из периодов;
- q_i — максимальная доля рискованного актива i в портфеле.

Ожидаемые переменные:

- x^b_{i1} — сумма, потраченная в период 1 на покупку актива i;
- x^s_{i1} — сумма, полученная в период 1 от продажи актива i.

Адаптивные переменные:

- $x_{it} = x_i(\omega^1, \dots, \omega^t)$ — сумма, инвестированная в актив i в период t, $t = 1, \dots, T$;
- $z_t = z_i(\omega^1, \dots, \omega^t)$ — сумма наличности в конце периода t, $t = 1, \dots, T$;
- $x^b_{it} = x^b_i(\omega^1, \dots, \omega^{t-1})$ — сумма, потраченная в период t на покупку актива i, $i = 1, \dots, n$, $t = 2, \dots, T$;
- $x^s_{it} = x^s_i(\omega^1, \dots, \omega^{t-1})$ — сумма, полученная в период t от продажи актива i, $i = 1, \dots, n$, $t = 2, \dots, T$;
- $\xi = \xi(\omega^1, \dots, \omega^T)$ — случайная составляющая стоимости портфеля в конце планового горизонта (в конце периода T);
- w — постоянная (безрисковая) составляющая стоимости портфеля в конце планового горизонта.

В выбранных переменных задача оптимизации портфеля формулируется следующим образом:

$$\lambda w + (1-\lambda)E(\xi) \to \max, \tag{8.11a}$$

$$z_{t-1} + \sum_{i=1}^{n}(1-\rho_{it})x^s_{it} - \sum_{i=1}^{n}(1+\rho_{it})x^b_{it} = \frac{1}{R}\, z_t, \quad t = 1, \dots, T, \tag{8.11b}$$

$$x_{i,t-1} + x^b_{it} - x^s_{it} = \frac{1}{r_{it}}\, x_{it}, \quad i = 1, \dots, n;\, t = 1, \dots, T, \tag{8.11c}$$

$$x_{it} - q_i\left(z_t + \sum_{j=1}^{n} x_{jt}\right) \le 0, \quad i = 1, \dots, n;\, t = 1, \dots, T, \tag{8.11d}$$

$$z_T + \sum_{i=1}^{n}(1-\rho_{iT})x_{iT} = w + \xi, \tag{8.11e}$$

$$x^b_{it}, x^s_{it}, x_{it} \ge 0, \quad i = 1, \dots, n;\, t = 1, \dots, T, \tag{8.11f}$$

$$z_t \ge 0, \quad t = 1, \dots, T. \tag{8.11g}$$

Целевая функция (8.11a) есть взвешенная ($0 \le \lambda \le 1$) комбинация двух величин: безрисковой (в наихудшем случае) w и ожидаемой (по всем возможным исходам) $E(\xi)$ стоимостей портфеля. Равенства (8.11b) задают баланс наличности в каждом периоде t: умноженная на процент на капитал сумма наличности в начале периода должна равняться сумме наличности z_t в конце периода. Сумма наличности в начале периода образуется как сумма наличности в конце предыдущего периода z_{t-1} плюс сумма, полученная от продажи рискованных активов $\sum_{i=1}^{n}(1-\rho_{it})x_{it}^{s}$, минус сумма, потраченная на покупку рискованных активов $\sum_{i=1}^{n}(1+\rho_{it})x_{it}^{b}$. Аналогично, равенства (8.11c) задают баланс для каждого актива i в каждом периоде t: умноженная на величину возврата сумма, инвестированная в актив в начале периода, равняется сумме x_{it}, инвестированной в актив в конце периода. Сумма, инвестированная в актив в начале периода, образуется как сумма $x_{i,t-1}$, инвестированная в актив в конце предшествующего периода, плюс сумма x_{it}^{b}, инвестированная в актив в данном периоде, минус сумма x_{it}^{s}, полученная от продажи актива в данный период. Неравенства (8.11d) ограничивают долю в портфеле каждого из рискованных активов. И наконец, равенство (8.11e) выделяет постоянную (безрисковую) составляющую стоимости портфеля в конце планового горизонта.

Пример 8.1. *Инвестор, обладающий суммой z_0, хочет часть этой суммы инвестировать в один рискованный актив. Плановый горизонт состоит из $T = 2$ периодов. Как и для общей модели, пусть R обозначает процент на капитал за один период. Для рискованного актива в период 1 с равной вероятностью 0.5 возврат равен r_1^+ или r_1^-, а в период 2 с вероятностью 2/3 возврат равен r_2^+ и с вероятностью 1/3 возврат равен r_2^-. Стоимость транзакции при покупке и продаже рискованного актива постоянна и равна ρ. Нужно записать детерминированный эквивалент для данной стохастической задачи.*

Решение. По условию, в начале планового горизонта инвестор имеет сумму z_0, а инвестиции в рискованный актив равны $x_0 = 0$. Дерево сценариев для данного примера представлено на рис. 8.2. В нем 7 узлов. Каждый из сценариев 1 и 2 стадии 1 случается с вероятностью 1/2, сценарии 3 и 5 случаются с вероятностью 1/3, а сценарии 4 и 6 — с вероятностью 1/6. Решение, принимаемое в узле i ($i = 0, 1, 2$), представляется переменными:

- x_i^b — сумма, потраченная на покупку рискованного актива в узле i;

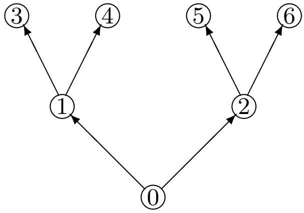


Рис. 8.2. Дерево сценариев для примера 8.1

- x_i^s — сумма, полученная от продажи рискованного актива в узле i.

В узлах $i = 1, \ldots, 6$ также вводятся следующие вспомогательные переменные:

- x_i — сумма, вложенная в рискованный актив в узле i;
- z_i — сумма наличности в узле i.

Заметим, что только переменные x_0^b и x_0^s являются ожидаемыми, а остальные — адаптивными. В этих переменных дискретный эквивалент модели (8.11) записывается следующим образом:

$$\lambda \cdot w + (1-\lambda)\left(\frac{1}{3}\xi_3 + \frac{1}{6}\xi_4 + \frac{1}{3}\xi_5 + \frac{1}{6}\xi_6\right) \to \max,$$

$$z_0 + (1-\rho)x_0^s - (1+\rho)x_0^b = \frac{1}{R}\, z_1, \qquad \text{узел 1}$$

$$x_0 + x_0^b - x_0^s = \frac{1}{r_1^+}\, x_1,$$

$$z_0 + (1-\rho)x_0^s - (1+\rho)x_0^b = \frac{1}{R} z_2, \qquad \text{узел 2}$$

$$x_0 + x_0^b - x_0^s = \frac{1}{r_1^-}\, x_2,$$

$$z_1 + (1-\rho)x_1^s - (1+\rho)x_1^b = \frac{1}{R}\, z_3, \qquad \text{узел 3}$$

$$x_1 + x_1^b - x_1^s = \frac{1}{r_2^+}\, x_3,$$

$$z_1 + (1-\rho)x_1^s - (1+\rho)x_1^b = \frac{1}{R}\, z_4, \qquad \text{узел 4}$$

$$x_1 + x_1^b - x_1^s = \frac{1}{r_2^-}\, x_4,$$

$$z_2 + (1-\rho)x_2^s - (1+\rho)x_2^b = \frac{1}{R}\, z_5, \qquad \text{узел 5}$$

$$
\begin{aligned}
x_2 + x_2^b - x_2^s &= \frac{1}{r_2^+}\, x_5, \\
z_2 + (1-\rho)x_2^s - (1+\rho)x_2^b &= \frac{1}{R}\, z_6, && \text{узел 6} \\
x_2 + x_2^b - x_2^s &= \frac{1}{r_2^-}\, x_6, \\
z_3 + (1-\rho)x_3 &= w + \xi_3, && \text{выделение} \\
z_4 + (1-\rho)x_4 &= w + \xi_4, && \text{постоянной} \\
z_5 + (1-\rho)x_5 &= w + \xi_5, && \text{составляющей} \\
z_6 + (1-\rho)x_6 &= w + \xi_6, && \text{стоимости} \\
x_1, z_1, x_2, z_2, x_3, z_3, x_4, z_4, x_5, z_5, x_6, z_6 &\ge 0, \\
x_0^b, x_0^s, x_1^b, x_1^s, x_2^b, x_2^s &\ge 0, \\
\xi_3, \xi_4, \xi_5, \xi_6 &\ge 0.
\end{aligned}
$$

□

8.3.2. Управление доходами

Управление доходами (*yield management*) — это методика для максимизации прибыли авиакомпаний, отелей и ряда других сервисных фирм со следующими характеристиками.

1. *Фиксированные издержки существенно больше переменных издержек.* Организовать авиарейс значительно дороже стоимости одного авиабилета.
2. *Возможность делить клиентов на категории* в зависимости от качества предоставляемых услуг.
3. *Исчезающая со временем выгода.* После вылета самолета исчезает и потенциальная прибыль от незанятых пассажирами мест в самолете.
4. *Возможность резервирования.* Авиакомпания может начать продавать билеты задолго до даты вылета самолета. При этом менеджеры компании должны определить стоимости билетов для каждой категории пассажиров в зависимости от времени, оставшегося до даты вылета. Поскольку стоимость билета растет с приближением даты вылета самолета, то менеджеры должны также решить,

сколько билетов каждой категории нужно продавать накануне вылета, а сколько в более ранние периоды.

5. *Изменчивость спроса.* Чтобы увеличить загрузку самолетов в периоды, когда пассажиропоток небольшой, авиакомпании снижают цены на билеты, а когда пассажиропоток увеличивается, то цены растут.

Теперь перейдем к конкретной постановке. Авиакомпания начинает продажу билетов на некоторый маршрут за D дней до заданной даты вылета. Плановый горизонт из D дней разделен на T периодов неравной продолжительности (например, плановый горизонт из $D = 60$ дней можно разбить на $T = 4$ периода продолжительностью 30, 20, 7 и 3 дня). На данном маршруте авиакомпания может использовать до k одинаковых самолетов, каждый из которых имеет q_1 мест первого класса (класса 1), q_2 — мест бизнес-класса (класса 2) и q_3 — мест эконом-класса (класса 3). До r_i процентов мест класса i могут быть трансформированы в места смежных классов, $i = 1, 2, 3$. Стоимость одного рейса равна f.

Цена билета класса i ($i = 1, 2, 3$) в период t ($t = 1, \dots, T$) может принимать одно из следующих значений $c_{ti1}, \dots, c_{t,i,O_t}$, где O_t есть количество *ценовых опций* в период t.

Спрос на билеты изменчив во времени и также зависит от цены билетов. Используя методы прогноза, для каждого периода t было выделено S_t сценариев. При этом сценарий s ($1 \le s \le S_t$) реализуется с вероятностью p_{ts}, $\sum_{s=1}^{S_t} p_{ts} = 1$, и в этом случае при использовании ценовой опции o спрос на билеты класса i равен d_{tsio}.

Нужно определить, сколько билетов каждого из классов и по какой цене продавать в каждый из периодов, чтобы максимизировать ожидаемую прибыль.

Для того чтобы записать детерминированную модель данной стохастической задачи, нужно сначала описать дерево сценариев. В данном приложении дерево сценариев имеет $n + 1 = 1 + \prod_{t=1}^{T} S_t$ узлов. Обозначим через V_t множество узлов уровня t, $t = 0, 1, \dots, T$. Будем считать, что корень дерева сценариев имеет номер 0 и тогда $V_0 = \{0\}$. Каждый узел $j \in V_t$ ($t = 1, \dots, T$) представляет одну из ситуаций, которая может сложиться после t периодов, и характеризуется набором чисел $(s_1, s_2, \dots, s_t)$, где $s_\tau \in \{1, \dots, S_t\}$ означает номер сценария, который реализовался в период τ. Ситуация, представляемая данным узлом j, реализуется с вероятностью $p_j \stackrel{\text{def}}{=} \prod_{\tau=1}^{t} p_{\tau,s_\tau}$ и при использовании ценовой опции o спрос на билеты класса i будет $\bar{d}_{jio} \stackrel{\text{def}}{=} d_{t,s_t,i,o}$, а стоимость этих билетов равна c_{tio}. Предком $parent(j)$ узла j является узел из V_{t-1}, который характеризуется набором чисел $(s_1, s_2, \dots, s_{t-1})$. Отметим, что предком всех узлов из V_1 является узел 0.

Теперь мы определим переменные. Пусть v обозначает количество используемых самолетов (количество рейсов). Каждому узлу $j \in V_{t-1}$, $t = 1, \dots, T$, поставим в соответствие следующие переменные:

- x_{jio} — количество билетов класса i, которые нужно продать в период t при выбранной ценовой опции o, когда после $(t-1)$-го периода реализуется ситуация, представляемая узлом j;
- $y_{jio} = 1$, если в период t для класса i применяется ценовая опция o, когда после $(t-1)$-го периода реализуется ситуация, представляемая узлом j, и $y_{jio} = 0$ в противном случае.

Каждому узлу $j \in V_t$, $t = 1, \dots, T$, припишем переменные:

- z_{ji} — количество билетов класса i, проданных за t периодов, предшествующих ситуации, представленной узлом j.

Теперь мы можем записать детерминированную модель следующим образом:

$$-fv + \sum_{t=1}^{T} \sum_{j \in V_{t-1}} \sum_{o=1}^{O_t} (\bar{p}_j c_{tio}) x_{jio} \to \max, \tag{8.12a}$$

$$\sum_{o=1}^{O_t} y_{jio} = 1, \quad j \in V_t;\ i = 1,2,3;\ t = 0,\dots,T-1, \tag{8.12b}$$

$$x_{jio} \le \bar{d}_{jio} y_{jio}, \quad j \in V_t;\ i = 1,2,3;\ o = 1,\dots,O_t;\ t = 0,\dots,T-1, \tag{8.12c}$$

$$z_{ji} = \sum_{o=1}^{O_1} x_{0io}\,, \quad j \in V_1;\ i = 1,2,3, \tag{8.12d}$$

$$z_{ji} = z_{parent(j),i} + \sum_{o=1}^{O_t} x_{parent(j),i,o}\,, \quad j \in V_t;\ i = 1,2,3;\ t = 2,\dots,T, \tag{8.12e}$$

$$z_{j1} \le (q_1 + \lfloor r_2 q_2/100 \rfloor)\, v, \quad j \in V_T, \tag{8.12f}$$

$$z_{j2} \le (q_2 + \lfloor (r_1 q_1 + r_3 q_3)/100 \rfloor)\, v, \quad j \in V_T, \tag{8.12g}$$

$$z_{j3} \le (q_3 + \lfloor r_2 q_2/100 \rfloor)\, v, \quad j \in V_T, \tag{8.12h}$$

$$z_{j1} + z_{j3} \le (q_1 + q_3 + \lfloor r_2 q_2/100 \rfloor)\, v, \quad j \in V_T, \tag{8.12i}$$

$$z_{j1} + z_{j2} + z_{j3} \le (q_1 + q_2 + q_3)\, v, \quad j \in V_T, \tag{8.12j}$$

$$x_{jio} \in \mathbb{Z}_+, \quad j \in V_t;\ i = 1,2,3;\ o = 1,\dots,O_t;\ t = 0,\dots,T-1, \tag{8.12k}$$
$$y_{jio} \in \{0,1\}, \quad j \in V_t;\ i = 1,2,3;\ o = 1,\dots,O_t;\ t = 0,\dots,T-1, \tag{8.12l}$$
$$z_{ji} \in \mathbb{Z}_+, \quad j \in V_t;\ i = 1,2,3;\ t = 1,\dots,T, \tag{8.12m}$$
$$v \in \mathbb{Z}_+\,,\ v \le k. \tag{8.12n}$$

Равенства (8.12b) обеспечивают то, что в любой из T периодов для каждого класса выбирается только одна ценовая опция. Переменные верхние границы (8.12c) гарантируют, что в любой период количество проданных билетов каждого класса не превосходит спроса на эти билеты для выбранных ценовых опций. Балансовые равенства (8.12d) и (8.12e) подсчитывают общее количество билетов каждого класса, проданных по завершению любого из T периодов. Неравенства (8.12f)–(8.12j) требуют, чтобы общее количество проданных билетов не превосходило количества мест в используемых самолетах.

Решив задачу (8.12), мы определим, какие ценовые опции для каждого из трех классов нужно использовать в первый период и сколько билетов каждого из классов нуждо продать в этот период (это определяется по значениям переменных x_{0io}). По завершению периода 1 будет известно количество билетов каждого из классов, проданных в данный период, и после этого можно будет записать новую модель для периодов $2,\dots,T$, чтобы заново определить, сколько билетов каждого из классов и по какой цене продавать в период 2. Такая процедура принятия решений повторяется и для остальных периодов $t = 3,\dots,T$.

8.4. Упражнения

8.1. Вы хотите инвестировать \$50 000. Акции XYZ сегодня продаются по \$20 за одну акцию. Европейский опцион стоимостью \$700 дает право (но не обязывает) через шесть месяцев купить 100 акций XYZ по цене \$15 за одну акцию. Кроме этого, шестимесячные безрисковые облигации номиналом \$100 сегодня продаются за \$90. Вы решили не покупать более 20 опционов.

Через шесть месяцев возможны три равновероятных сценария для цены акции XYZ: 1) цена не изменится; 2) цена вырастет до \$40; 3) цена упадет до \$12.

8.2. Докажите, что при фиксированном x функция $f(x,\omega)$, определенная по формуле (8.3), в действительности является случайной величиной.

Сформулируйте и решите задачи СЦП, в которых вы хотите сформировать портфель с целью максимизировать:

а) ожидаемый доход;

б) ожидаемый доход, при условии, что в любом из трех сценариев доход не должен быть меньше \$2000;

в) *безрисковый доход*, который определяется равным доходу в наихудшем из трех возможных сценариев.

Сравните оптимальные решения для трех моделей.

8.3. Фирма, производящая майки со специальной символикой, накануне очередного мероприятия (фестиваля, спортивного соревнования и т. д.) должна решить, сколько маек нужно произвести. В дни проведения мероприятия фирма может продать майки по цене \$20 за майку. Но после завершения мероприятия нераспроданные майки можно продать только по цене \$4 за майку. Стоимость производства одной специальной майки \$8. Фирма оценивает спрос на майки во время данного мероприятия следующим образом:

Спрос	Вероятность
300	0.05
400	0.1
500	0.4
600	0.3
700	0.1
800	0.05

Сколько маек нужно произвести, чтобы ожидаемая прибыль была максимальной?

8.4. Фирма хочет потратить \$4 000 на производство трех делимых продуктов. Издержки производства единицы продукта 1 равны \$1, продукта 2 — \$2, продукта 3 — \$4. Цена единицы продукта 1 фиксирована и равна \$1.20, продукта 2 — \$2.30, продукта 3 — \$4.50. Спрос на эти продукты есть независимые случайные величины, равномерно распределенные на отрезке $[0, 3\,000]$. Сколько единиц каждого из продуктов нужно произвести, чтобы максимизировать ожидаемую прибыль.

Глава 9

Теория массового обслуживания

Каждая *система массового обслуживания* (СМО) состоит из одного или нескольких «приборов», которые мы будем называть *каналами* обслуживания. Каналами могут быть: линии связи, билетные кассы, лифты, такси, вебсерверы, серверы баз данных и др. СМО могут быть *одноканальными* и *многоканальными*.

Всякая СМО предназначена для обслуживания некоторого потока заявок (или «требований»), которые поступают в случайные моменты времени. Обслуживание заявки продолжается какое-то время (в общем случае продолжительность обслуживания заявки есть случайная величина), после чего канал освобождается и готов к обслуживанию следующей заявки. Случайный характер потока заявок и продолжительности их обслуживания приводит к тому, что в некоторые периоды времени на входе СМО может скапливаться излишне большое число заявок (они либо становятся в очередь, либо покидают СМО необслуженными); в другие же периоды отдельные каналы СМО могут простаивать.

Процесс работы СМО представляет собой случайный процесс с дискретными состояниями и непрерывным временем; состояние СМО меняется скачком в моменты, когда появляется новая заявка, или завершается обслуживание некоторой заявки, или заявка, которой надоело ждать в очереди, покидает очередь.

В дальнейшем, если это не оговорено особо, мы будем предполагать, что все потоки заявок и обслуживаний являются *пуассоновскими*.

9.1. Потоки событий

Бесконечное семейство случайных величин $\{X(t)\}_{t\in\mathbb{R}_+}$ называется *пуассоновским процессом с параметром* (или *средним*) λ, если оно удовлетворяет следующим условиям:

(i) $X(0) = 0$;

(ii) (*отсутствие памяти*) приращения $X(\tau_i + t_i) - X(\tau_i)$ на произвольных непересекающихся интервалах $[\tau_i, \tau_i + t_i]$, $i = 1, \ldots, k$, — независимые случайные величины;

(iii) (*стационарность*) для любого $t \in \mathbb{R}_+$ случайная величина имеет распределение Пуассона $\pi_{\lambda t}$[11].

В курсах по теории вероятностей доказывается (см., например, [12]), что любой пуассоновский процесс можно проинтерпретировать следующим образом. Рассмотрим бесконечную последовательность случайных величин $T_1, T_2, \ldots$, имеющих экспоненциальную плотность распределения $\lambda e^{-\lambda t}$, $\lambda > 0$. Например, $T_1, T_2, \ldots$ могут быть интервалы времени между последовательными событиями некоторого *потока событий*, каким может быть поток автомобилей на некотором перекрестке, поток покупателей у кассы в супермаркете, поток вызовов скорой помощи, поток отказов некоторого технического устройства, поток запросов информации с некоторого вебсервера и т. д..

Поскольку средняя продолжительность интервала между последовательными событиями $E(T_j) = 1/\lambda$, то параметр λ можно рассматривать как *интенсивность потока*, которая равна среднему количеству событий, происходящих в единицу времени.

Обозначим через $N(t)$ количество событий, произошедших в промежутке времени $[0, t]$. Можно доказать (см., например, [12]), что семейство $\{N(t)\}_{t\in\mathbb{R}_+}$ является пуассоновским процессом с параметром λ. В частности,

$$\mathbb{P}(N(t) = k) = \mathbb{P}(T_1 + T_2 + \cdots + T_k \le t) = \pi_{\lambda t}(k), \quad k = 0, 1, 2, \ldots$$

[11] Дискретная случайная величина Y, принимающая неотрицательные целые значения, имеет распределение Пуассона π_α с параметром α, $\mathbb{P}(Y = k) = \pi_\alpha(k) \stackrel{\text{def}}{=} \frac{e^{-\alpha}}{k!}\alpha^k$ для всех $k \in \mathbb{Z}_+$.

9.2. Схема гибели и размножения

Термин «схема гибели и размножения» в биологии описывает изменение численности популяции. Схема гибели и размножения очень часто встречается и в задачах теории массового обслуживания, поэтому мы и начинаем с ее рассмотрения. Граф состояний для схемы гибели и размножения имеет вид, показанный на рис. 9.1.

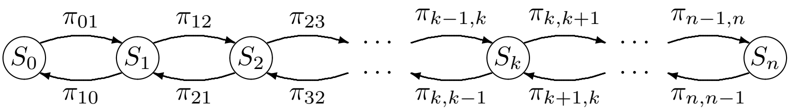


Рис. 9.1. Граф состояний для схемы гибели и размножения

9.2.1. Уравнения Колмогорова

Обозначим через $p_k(t)$ вероятность того, что в момент времени t система находится в состоянии S_k. Для достаточно малого $\Delta t > 0$ в момент времени $t + \Delta t$ система окажется в состоянии S_k $(1 < k < n)$

- с вероятностью $\pi_{k-1,k}\,\Delta t$, если в момент t она была в состоянии S_{k-1};
- с вероятностью $1 - (\pi_{k,k-1} + \pi_{k,k+1})\Delta t$, если в момент t она была в состоянии S_k;
- с вероятностью $\pi_{k+1,k}\,\Delta t$, если в момент t она была в состоянии S_{k+1}.

Поэтому справедливо равенство

$$\begin{aligned} p_k(t + \Delta t) = p_{k+1}(t)\pi_{k+1,k}\Delta t + p_{k-1}(t)\pi_{k-1,k}\Delta t \\ + p_k(t)\left(1 - (\pi_{k,k+1} + \pi_{k,k-1})\Delta t\right). \end{aligned}$$

Разделив обе части равенства на Δt, получим

$$\frac{p_k(t+\Delta t) - p_k(t)}{\Delta t} = \pi_{k+1,k}p_{k+1}(t) + \pi_{k-1,k}p_{k-1}(t) - (\pi_{k,k+1} + \pi_{k,k-1})p_k(t).$$

Переходя к пределу при $\Delta t \to 0$, получим

$$\begin{aligned} \frac{dp_k(t)}{dt} &= \pi_{k+1,k}p_{k+1}(t) + \pi_{k-1,k}p_{k-1}(t) \\ &- (\pi_{k,k+1} + \pi_{k,k-1})p_k(t), \quad k = 1, \ldots, n-1. \end{aligned} \tag{9.1}$$

Аналогично, можно получить уравнения для $k = 0$ и $k = n$:

$$\frac{dp_0(t)}{dt} = \pi_{10}p_1(t) - \pi_{01}p_0(t), \tag{9.2}$$
$$\frac{dp_n(t)}{dt} = \pi_{n-1,n}p_{n-1}(t) - \pi_{n,n-1}p_n(t). \tag{9.3}$$

Если в системе установился стационарный режим, то все вероятности $p_k(t) \stackrel{\text{def}}{=} p_k$ постоянны (независят от времени). Мы можем вычислить *финальные вероятности* $p_0, p_1, \ldots, p_n$[12] состояний системы, решая систему уравнений (9.1)–(9.3), с учетом того, что $\frac{dp_k(t)}{dt} = 0$ для $k = 0, 1, \ldots, n$. Для состояния S_0 справедливо равенство:

$$\pi_{01}p_0 = \pi_{10}p_1. \tag{9.4}$$

Для состояния S_1 имеем:

$$(\pi_{10} + \pi_{12})p_1 = \pi_{01}p_0 + \pi_{21}p_2.$$

В силу (9.4) последнее равенство приводится к виду

$$\pi_{12}p_1 = \pi_{21}p_2.$$

Далее, совершенно аналогично получаем равенство

$$\pi_{23}p_2 = \pi_{32}p_3$$

и для любого $k = 1, \ldots, n$ имеем:

$$\pi_{k-1,k}p_{k-1} = \pi_{k,k-1}p_k.$$

Итак, финальные вероятности $p_0, p_1, \ldots, p_n$ удовлетворяют системе

$$\begin{aligned}
\pi_{01}p_0 &= \pi_{10}p_1, \\
\pi_{12}p_1 &= \pi_{21}p_2, \\
&\ldots \\
\pi_{k-1,k}p_{k-1} &= \pi_{k,k-1}p_k, \\
&\ldots \\
\pi_{n-1,n}p_{n-1} &= \pi_{n,n-1}p_n.
\end{aligned} \tag{9.5}$$

[12] Мы можем интерперетировать p_i как долю времени, когда система пребывает в состоянии S_i.

Из первого уравнения системы (9.5) вызазим p_1 через p_0:

$$p_1 = \frac{\pi_{01}}{\pi_{10}} p_0. \tag{9.6}$$

Из второго, с учетом (9.6), найдем:

$$p_2 = \frac{\pi_{12}}{\pi_{21}} p_1 = \frac{\pi_{01}\pi_{12}}{\pi_{21}\pi_{10}} p_0. \tag{9.7}$$

Из третьего, с учетом (9.7), получим:

$$p_3 = \frac{\pi_{23}}{\pi_{32}} p_2 = \frac{\pi_{01}\pi_{12}\pi_{23}}{\pi_{32}\pi_{21}\pi_{10}} p_0. \tag{9.8}$$

В общем, для любого $k = 1, \ldots, n$ имеем:

$$p_k = \frac{\pi_{01}\pi_{12} \ldots \pi_{k-1,k}}{\pi_{k,k-1} \ldots \pi_{21}\pi_{10}} p_0. \tag{9.9}$$

Заметим, что в формуле (9.9) числитель есть произведение всех интенсивностей, стоящих у дуг, ведущих слева направо от состояния S_0 до состояния S_k, а знаменатель есть произведение всех интенсивностей, стоящих у дуг, ведущих справа налево от состояния S_k до состояния S_0.

Таким образом, все вероятности состояний $p_1, \ldots, p_n$ выражаются через состояние p_0. Подставив эти выражения в нормировочное равенство

$$p_0 + p_1 + p_2 + \cdots + p_n = 1,$$

найдем

$$p_0 = \left(1 + \frac{\pi_{01}}{\pi_{10}} + \frac{\pi_{01}\pi_{12}}{\pi_{21}\pi_{10}} + \cdots + \frac{\pi_{01}\pi_{12} \ldots \pi_{n-1,n}}{\pi_{n,n-1} \ldots \pi_{21}\pi_{10}}\right)^{-1}. \tag{9.10}$$

9.3. Формулы Литтла

В этом параграфе мы выведем важную формулу, связывающую (для предельного стационарного режима) среднее число заявок $L_{\text{сист}}$, находящихся в СМО (т. е. обслуживаемых или стоящих в очереди), и среднее время пребывания заявки в системе $W_{\text{сист}}$.

Рассмотрим любую СМО (одноканальную или многоканальную, марковскую или немарковскую, с неограниченной или с ограниченной очередью, и т. д.) и связанные с нею два потока событий: поток заявок,

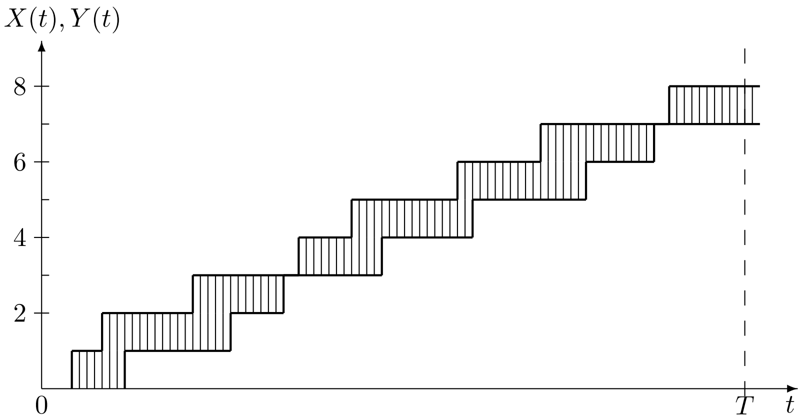


Рис. 9.2. Поведение функций $X(t)$ и $Y(t)$

поступающих в СМО, и поток заявок, *покидающих* СМО. Если в системе установился предельный стационарный режим, то среднее число заявок, поступаюших в СМО, равно среднему числу заявок, покидающих СМО, т. е. оба потока имеют одну и ту же интенсивность λ.

Обозначим через $X(t)$ число заявок, поступивших в СМО до момента времени t, а через $Y(t)$ число заявок, покинувших СМО до момента t. И та и другая функции являются случайными, $X(t)$ увеличиваются на единицу в момент поступления новой заявки, а $Y(t)$ уменьшается на единицу в момент, когда некоторая заявка покидает систему. Поведение функций $X(t)$ и $Y(t)$ проиллюстрировано на рис. 9.2. Для любого момента t разность $Z(t) = X(t) - Y(t)$ есть число заявок, находящихся в СМО. Когда $Z(t) = 0$, в системе нет заявок.

Рассмотрим очень большой промежуток времени T и вычислим для него среднее число заявок, находящихся в СМО. Оно будет равно

$$\frac{1}{T}\int_0^T Z(t)dt.$$

Этот интеграл равен площади фигуры, заштрихованной на рис. 9.2. Фигура состоит из прямоугольников, k-й из которых имеет высоту, равную единице, и основание, равное времени t_k пребывания в системе заявки, поступившей k-й по счету. Отметим, что в конце промежутка T некоторые прямоугольники войдут в заштрихованную фигуру не полностью, а частично, но при достаточно больших T

$$\frac{1}{T}\int_0^T Z(t)dt \approx \sum_{k=1}^{k(T)} t_k,$$

где $k(T)$ обозначает количество заявок, поступивших в систему за время T.

Отсюда получаем

$$L_{\text{сист}} = \lim_{T\to\infty}\frac{1}{T}\int_0^T Z(t)dt = \lim_{T\to\infty}\frac{1}{T}\sum_{k=1}^{k(T)} t_k = \lambda \lim_{T\to\infty}\frac{1}{T\lambda}\sum_{k=1}^{k(T)} t_k.$$

Но величина $T\lambda$ есть среднее число заявок, поступивших за время T. Поэтому

$$\lim_{T\to\infty}\frac{1}{T\lambda}\sum_{k=1}^{k(T)} t_k$$

есть среднее время пребывания заявки в системе $W_{\text{сист}}$. Итак $L_{\text{сист}} = \lambda W_{\text{сист}}$, или

$$W_{\text{сист}} = \frac{1}{\lambda}L_{\text{сист}}. \tag{9.11}$$

Это и есть *первая формула Литтла*: для любой СМО, при любом характере потока заявок, при любом распределении времени обслуживания, при любой дисциплине обслуживания *среднее время пребывания заявки в системе равно среднему числу заявок в системе, деленному на интенсивность потока заявок.*

Точно таким же образом выводится *вторая формула Литтла*, связывающая среднее время пребывания заявки в очереди $W_{\text{оч}}$ и среднее число заявок в очереди $L_{\text{оч}}$:

$$W_{\text{оч}} = \frac{1}{\lambda}L_{\text{оч}}. \tag{9.12}$$

Для вывода формулы (9.12) достаточно заменить функцию Y на функцию U, где $U(t)$ есть количество заявок, покинувших очередь до момента t (если заявка, поступая в систему, обслуживается сразу, не становясь в очередь, то можно считать, что она пробыла в очереди нулевое время).

9.4. Многоканальная СМО с отказами

Здесь мы рассмотрим одну из первых по времени «классических» задач теории массового обслуживания. Эта задача возникла из практических нужд телефонии и была решена в начале 19-го века датским математиком Эрлангом.

Имеется n каналов (линий связи), на которые поступает поток заявок с интенсивностью λ. Поток обслуживаний имеет интенсивность μ. Нужно найти финальные вероятности состояний СМО, а также характеристики ее эффективности:

- A — абсолютную пропускную способность, т. е. среднее число заявок, обслуживаемых в единицу времени;
- Q — относительную пропускную способность, т. е. среднею долю пришедших заявок, обслуженных системой;
- $P_{\text{отк}}$ — вероятность отказа, т. е. того, что заявка покинет СМО необслуженной;
- $\bar{k}$ — среднее число занятых каналов.

Состояние данной СМО определяется числом заявок в системе (в данном случае оно совпадает с числом занятых каналов): S_k — в СМО находится k заявок ($k = 1, \ldots, n$). Граф состояний СМО соответствует схеме гибели и размножения и представлен на рис. 9.3.

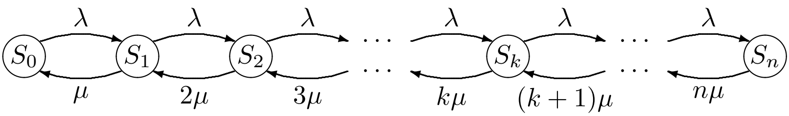


Рис. 9.3. Граф состояний n-канальной СМО с отказами

А теперь воспользуемся формулами (9.9) и (9.10) для финальных вероятностей в схеме гибели и размножения. По формуле (9.10) получим:

$$p_0 = \left(1 + \frac{\lambda}{\mu} + \frac{\lambda^2}{2\mu^2} + \frac{\lambda^3}{3!\mu^3} + \cdots + \frac{\lambda^k}{k!\mu^k} + \ldots \frac{\lambda^n}{n!\mu^n}\right)^{-1}. \tag{9.13}$$

Члены разложения $\frac{\lambda}{\mu}, \frac{\lambda^2}{2\mu}, \ldots, \frac{\lambda^n}{n!\mu^n}$ являются коэффициентами при p_0 в выражениях для $p_1, \ldots, p_n$:

$$p_k = \frac{\lambda^k}{k!\mu^k} p_0, \quad k = 1, \ldots, n. \tag{9.14}$$

Обозначим отношение λ/μ через ρ и назовем его «приведенной интенсивностью потока заявок». Заметим, что ρ есть среднее число заявок, приходящее за среднее время обслуживания одной заявки. Пользуясь этим обозначением, перепишем формулы (9.13) и (9.14) следующим образом:

$$p_0 = \left(\sum_{k=0}^{n} \frac{\rho^k}{k!}\right)^{-1}, \tag{9.15}$$

$$p_k = \frac{\rho^k}{k!} p_0, \quad k = 1, \ldots, k. \tag{9.16}$$

Эти формулы известны как формулы Эрланга.

Теперь мы можем вычислить характеристики эффективности СМО. Вероятность того, что пришедшая заявка получит отказ (не будет обслужена) равна

$$P_{\text{отк}} = p_n = \frac{\rho^n}{n!} p_0.$$

Далее находим относительную пропускную способность — вероятность того, что пришедшая заявка будет обслужена:

$$Q = 1 - P_{\text{отк}} = 1 - \frac{\rho^n}{n!} p_0.$$

Абсолютную пропускную способность получим, умножая интенсивность потока заявок λ на Q:

$$A = \lambda Q = \lambda \left(1 - \frac{\rho^n}{n!} p_0\right). \tag{9.17}$$

Осталось только найти среднее число занятых каналов $\bar{k}$. Поскольку

$$\bar{k} = 0 \cdot p_0 + 1 \cdot p_1 + 2 \cdot p_2 + \cdots + n \cdot p_n,$$

то мы могли бы вычислить $\bar{k}$, подставляя в эту формулу выражения (9.16) для p_k $(k = 1, \ldots, n)$ и выполняя соответствующие упрощения. Но мы получим выражение для $\bar{k}$ более простым способом. В самом деле, мы знаем интенсивность потока обслуженных системой заявок A. Каждый занятый канал в единицу времени обслуживает в среднем μ заявок. Следовательно, среднее число занятых каналов равно $\bar{k} = A/\mu$, или, учитывая (9.17),

$$\bar{k} = \rho \left(1 - \frac{\rho^n}{n!} p_0\right).$$

Пример 9.1. *Станция связи имеет три канала ($n = 3$), интенсивность потока заявок $\lambda = 1.5$ (заявки в минуту), среднее время обслуживания одной заявки 2 мин. ($\mu = 1/2 = 0.5$). Найти финальные вероятности состояний и характеристики эффективности СМО: A, Q, $P_{отк}$, $\bar{k}$. Сколько нужно каналов, чтобы удовлетворять не менее 80 % заявок? Какая доля каналов при этом будет простаивать?*

9.5. Одноканальная СМО с неограниченной очередью

Рассмотрим одноканальную СМО с очередью, на которую не наложено никаких ограничений (ни по длине очереди, ни по времени ожидания). В СМО поступает поток заявок интенсивности λ, а поток обслуживаний имеет интенсивность μ. Нужно найти финальные вероятности состояний СМО, а также характеристики ее эффективности:

- $L_{\text{сист}}$ — среднее число заявок в системе;
- $W_{\text{сист}}$ — среднее время пребывания заявки в системе;
- $L_{\text{оч}}$ — среднее число заявок в очереди;
- $W_{\text{оч}}$ — среднее время пребывания заявки в очереди;
- $P_{\text{зан}}$ — вероятность того, что канал занят (степень загрузки канала).

Как и ранее, состояние данной СМО определяется числом заявок в системе: S_k — в СМО находится k заявок ($k = 1, 2, \dots$). Граф состояний СМО соответствует системе гибели и размножения и представлен на рис. 9.4.

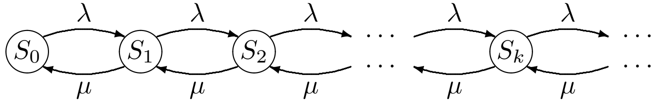


Рис. 9.4. Граф состояний n-канальной СМО с отказами

Поскольку число состояний в данной СМО бесконечно, то при $t \to \infty$ очередь может неограниченно возрастать. Поэтому финальные вероятности существуют не всегда, а только когда система не перегружена. Можно доказать, что если $\rho < 1$, то финальные вероятности существуют, а при $\rho \geq 1$ очередь при $t \to \infty$ растет неограниченно. Особенно «непонятным» кажется этот факт при $\rho = 1$. Казалось бы, к системе не предъявляется невыполнимых требований: за время обслуживания одной заявки приходит в среднем одна заявка, и все должно быть в порядке, а вот на деле — не так. При $\rho = 1$ СМО справляется с потоком заявок, только если поток этот — регулярен, и время обслуживания — тоже не случайное, равное интервалу между заявками. В этом «идеальном» случае очереди в СМО вообще не будет, канал будет непрерывно занят и будет регулярно выпускать обслуженные заявки. Но стоит только потоку заявок или потоку обслуживаний стать хотя бы чуточку случайным — и очередь уже будет расти до бесконечности. На практике этого не происходит только потому, что «бесконечное число заявок в очереди»— абстракция.

Вернемся к анализу нашей СМО. Формулы (9.9) и (9.10) для финальных вероятностей в схеме гибели и размножения были выведены только для случая конечного числа состояний. Мы позволим себе вольность — воспользуемся ими и для бесконечного числа состояний:

$$p_0 = \left(1 + \frac{\lambda}{\mu} + \frac{\lambda^2}{\mu^2} + \dots + \frac{\lambda^k}{\mu^k} + \dots\right)^{-1} = \\ = (1 + \rho + \rho^2 + \dots + \rho^k + \dots)^{-1}. \tag{9.18}$$

Ряд в формуле (9.18) представляет собой геометрическую прогрессию. При $\rho < 1$ ряд сходится — это бесконечно убывающая геометрическая прогрессия со знаменателем ρ. При $\rho \geq 1$ ряд расходится, что косвенно подтверждает то, что финальные вероятности состояний $p_0, p_1, \dots, p_k, \dots$ существуют только при $\rho < 1$.

При $\rho < 1$ из (9.18) имеем

$$p_0 = 1 - \rho. \tag{9.19}$$

Поскольку $p_k = \rho^k p_0$, то остальные вероятности $p_1, p_2, \dots, p_k, \dots$ определяются по формулам:

$$p_1 = \rho(1 - \rho),\ p_2 = \rho^2(1 - \rho),\ \dots,\ p_k = \rho^k(1 - \rho),\ \dots \tag{9.20}$$

Как ни странно, но, поскольку максимальная из этих вероятностей есть p_0, то наиболее вероятное число заявок в системе будет 0.

Найдем теперь среднее число заявок в системе. Случайная величина ξ — число заявок в системе — принимает значения $0, 1, 2, \dots, k, \dots$ с вероятностями $p_0, p_1, p_2, \ldots, p_k, \ldots$ Ее математическое ожидание равно

$$\begin{aligned}
L_{\text{сист}} &= \sum_{k=0}^{\infty} k p_k = \sum_{k=1}^{\infty} k\rho^k (1-\rho) = \rho(1-\rho) \sum_{k=1}^{\infty} k\rho^{k-1} \\
&= \rho(1-\rho) \sum_{k=1}^{\infty} \frac{d}{d\rho}\rho^k = \rho(1-\rho) \frac{d}{d\rho} \sum_{k=1}^{\infty} \rho^k \\
&= \rho(1-\rho) \frac{d}{d\rho} \frac{\rho}{1-\rho} = \rho(1-\rho) \frac{1}{(1-\rho)^2} \\
&= \frac{\rho}{1-\rho}.
\end{aligned}$$

Применим формулу Литтла (9.11) и найдем среднее время пребывания заявки в системе:

$$W_{\text{сист}} = \frac{\rho}{\lambda(1-\rho)}.$$

Теперь определим среднее число заявок в очереди. Число заявок в очереди равно числу заявок в системе минус число заявок, находящихся под обслуживанием. Значит (по правилу сложения математических ожиданий), среднее число заявок в очереди $L_{\text{оч}}$ равно среднему числу заявок в системе $L_{\text{сист}}$ минус среднее число заявок под обслуживанием. Число заявок под обслуживанием может быть либо нулем (если канал свободен), либо единицей (если канал занят). Математическое ожидание такой случайной величины равно вероятности того, что канал занят (мы ее обозначили $P_{\text{зан}}$). Ясно, что $P_{\text{зан}}$ равно единице минус вероятность p_0 того, что канал свободен:

$$P_{\text{зан}} = 1 - p_0 = \rho.$$

Следовательно, среднее число заявок под обслуживанием равно $L_{\text{об}} = \rho$, откуда

$$L_{\text{оч}} = L_{\text{сист}} - \rho = \frac{\rho}{1-\rho} - \rho = \frac{\rho^2}{1-\rho}. \tag{9.21}$$

По формуле Литтла (9.12) найдем среднее время пребывания заявки в очереди:

$$W_{\text{оч}} = \frac{\rho^2}{\lambda(1-\rho)}.$$

Пример 9.2. *Ресторан MacDonalds планирует открыть drive-through окно для обслуживания своих клиентов. Менеджеры оценили, что клиенты будут прибывать с интенсивностью* 15 *клиентов в час. Кассир,* *который будет работать в данном окне, в среднем тратит три минуты на обслуживание одного клиента. Нужно определить параметры эффективности данной СМО.*

Решение. Параметры данной СМО следующие: $\lambda = 15$, $\mu = 60/3 = 20$. Средняя занятость кассира $\rho = \lambda/\mu = 15/20 = 0.75$ (75 %).

1. Среднее число клиентов в системе:

$$L_{\text{сист}} = \frac{\rho}{1-\rho} = \frac{0.75}{1-0.75} = 3 \text{ клиента.}$$

2. Среднее число клиентов в очереди:

$$L_{\text{оч}} = L_{\text{сист}} - \rho = 3 - 0.75 = 2.25 \text{ клиента.}$$

3. Среднее время ожидания в системе:

$$W_{\text{сист}} = \frac{1}{\lambda} L_{\text{сист}} = \frac{3}{15} = 0.2 \text{ ч.} = 12 \text{ мин.}$$

4. Среднее время ожидания в очереди:

$$W_{\text{оч}} = \frac{1}{\lambda} L_{\text{оч}} = \frac{2.25}{15} = 0.15 \text{ ч.} = 9 \text{ мин.}$$

9.6. Многоканальная СМО с неограниченной очередью

Совершенно аналогично расчитываетя эффективность работы n-канальной СМО с неограниченной очередью. Нумерация состояний теперь следующая:

- S_k — занято k каналов, остальные свободны ($k = 0, \dots, n$);
- S_{n+r} — заняты все n каналов, r заявок стоит в очереди ($r = 1, 2, \dots$).

Граф состояний такой СМО представлен на рис. 9.5. Он представляет схему гибели и размножения, но с бесконечным числом состояний.

Примем без доказательства естественное условие существования финальных вероятностей: $\rho/n < 1$ (напомним, что $\rho = \lambda/\mu$). Если $\rho/n \geq 1$, то очередь растет до бесконечности.

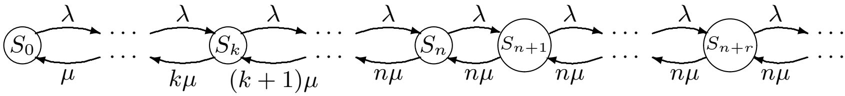


Рис. 9.5. Граф состояний n-канальной СМО с неограниченной очередью

Поэтому предположим, что $\rho/n < 1$ и финальные вероятности существуют. По формуде (9.10) найдем $1/p_0$:

$$\begin{aligned}\frac{1}{p_0} &= 1 + \frac{\rho}{1!} + \frac{\rho^2}{2!} + \dots + \frac{\rho^n}{n!} + \frac{\rho^{n+1}}{n \cdot n!} + \frac{\rho^{n+2}}{n^2 \cdot n!} + \frac{\rho^{n+3}}{n^3 \cdot n!} + \dots \\ &= 1 + \frac{\rho}{1!} + \frac{\rho^2}{2!} + \dots + \frac{\rho^n}{n!} + \frac{\rho^{n+1}}{n \cdot n!}\left(1 + \frac{\rho}{n} + \left(\frac{\rho}{n}\right)^2 + \dots\right) \\ &= 1 + \frac{\rho}{1!} + \frac{\rho^2}{2!} + \dots + \frac{\rho^n}{n!} + \frac{\rho^{n+1}}{n!(n-\rho)}.\end{aligned}$$

По формулам (9.9) найдем остльные вероятности:

$$p_0 = \left(1 + \frac{\rho}{1!} + \frac{\rho^2}{2!} + \dots + \frac{\rho^n}{n!} + \frac{\rho^{n+1}}{n!(n-\rho)}\right)^{-1}, \tag{9.22a}$$

$$p_1 = \frac{\rho}{1!}p_0, \ \dots, p_k = \frac{\rho^k}{k!}p_0, \ \dots, p_n = \frac{\rho^n}{n!}p_0, \tag{9.22b}$$

$$p_{n+1} = \frac{\rho^{n+1}}{n \cdot n!}p_0, \ \dots, p_{n+r} = \frac{\rho^{n+r}}{n^r \cdot n!}p_0, \ \dots \tag{9.22c}$$

Теперь найдем характеристики эффективности данной СМО. Среднее число занятых каналов для любой СМО с неограниченной очередью определяется одинаково: $\bar{k} = \lambda/\mu$.

Среднее число заявок в очереди вычисляется так:

$$\begin{aligned}L_{\text{оч}} &= \sum_{r=1}^{\infty} r p_{n+r} = \sum_{r=1}^{\infty} r \frac{\rho^{n+r}}{n^r \cdot n!}p_0 = \frac{\rho^{n+1}p_0}{n!}\sum_{r=1}^{\infty}\frac{r\rho^{r-1}}{n^r} = \\ &= \frac{\rho^{n+1}p_0}{n!}\sum_{r=1}^{\infty}\frac{d}{d\rho}\frac{\rho^r}{n^r} = \frac{\rho^{n+1}p_0}{n!}\frac{d}{d\rho}\sum_{r=1}^{\infty}\left(\frac{\rho}{n}\right)^r = \\ &= \frac{\rho^{n+1}p_0}{n!}\frac{d}{d\rho}\left(\frac{\rho/n}{1-\rho/n}\right) = \frac{\rho^{n+1}p_0}{n \cdot n!(1-\rho/n)^2}.\end{aligned} \tag{9.23}$$

Среднее число заявок в системе $L_{\text{сист}}$ равно среднему числу заявок в очереди $L_{\text{оч}}$ плюс число заявок под обслуживанием, которое равно среднему числу занятых каналов $\bar{k} = \rho$. Итак,

$$L_{\text{сист}} = L_{\text{оч}} + \rho.$$

И наконец, по формуле Литтла получим средние времена пребывания заявки в очереди и в системе:

$$W_{\text{оч}} = \frac{1}{\lambda} L_{\text{оч}}, \quad W_{\text{сист}} = \frac{1}{\lambda} L_{\text{сист}}.$$

Пример 9.3. *На автовоклале имеются всего две кассы: одна продает билеты на маршруты направления A, а другая — на маршруты направления B. Интенсивность потока заявок (пассажиров, желающих купить билеты) для обоих направлений одинакова: $\lambda_A = \lambda_B = 0.45$ (пассажира в минуту). Кассир тратит на обслуживания пассажира в среднем две минуты ($\mu_A = \mu_B = 0.5$). Определите среднюю длину очереди и среднее время ожидания в очереди для каждой из двух касс (одноканальных СМО с очередью). Как изменятся эти параметры эффективности, если две очереди объединить в одну и обе кассы начнут продавать билеты на оба напрвления?*

Решение. В настоящий момент мы имеем две одноканальных СМО; на каждую поступает поток заявок с интенсивностью $\lambda = 0.45$; интенсивность потока обслуживания $\mu = 0.5$. Поскольку $\rho = \lambda/\mu = 0.9 < 1$, то финальные вероятности существуют. По формуле (9.21) вычисляем среднюю длину очереди:

$$L_{\text{оч}} = \frac{\rho^2}{1-\rho} = \frac{0.9^2}{1-0.9} = 8.1.$$

Деля $L_{\text{оч}}$ на λ, найдем среднее время ожидания в очереди

$$W_{\text{оч}} = \frac{L_{\text{оч}}}{\lambda} = \frac{8.1}{0.45} \approx 18 \text{ (минут)}.$$

Теперь рассмотрим случай, когда обе кассы продают билеты на оба направления. На двухканальную СМО поступает поток заявок с интенсивностью $\lambda = \lambda_A + \lambda_B = 2 \cdot 0.45 = 0.9$. Интенсивность потока потока обслуживаний каждым каналом $\mu = 0.5$. Поэтому $\rho = \lambda/\mu = 1.8$. Поскольку $\rho/n = 1.8/2 = 0.9 < 1$, то финальные вероятности существуют.

По формуле (9.22a) находим

$$\begin{aligned} p_0 &= \left(1 + \frac{\rho}{1!} + \frac{\rho^2}{2!} + \frac{\rho^3}{2!(2-\rho)}\right)^{-1} \\ &= \left(1 + 1.8 + \frac{1.8^2}{2} + \frac{1.8^3}{2(2-1.8)}\right) -1 \approx 0.0525. \end{aligned}$$

Среднее число заявок в очереди находим по формуле (9.23):

$$L_{\text{оч}} = \frac{\rho^{n+1} p_0}{n \cdot n!(1-\rho/n)^2} = \frac{1.8^3 \cdot 0.0525}{2 \cdot 2(1-1.8/2)^2} \approx 7.68.$$

Деля $L_{\text{оч}}$ на λ, найдем среднее время ожидания в очереди

$$W_{\text{оч}} = \frac{L_{\text{оч}}}{\lambda} = \frac{7.68}{0.9} \approx 8.54 \text{ (минуты)}.$$

Почему произошло такое сокращение времени ожидания в очереди? Во первых, в двуканальной СМО меньше время простаивания каждого из двух кассиров. Объяснение этому следующее: при двух одноканальных СМО кассир, который обслужил очередного пассажира, будет простаивать, если в очереди нет пассажиров на его направление, а в двуканальной СМО кассир, который обслужил очередного пассажира, будет простаивать, если общая очередь пуста (нет пассажиров на оба направления).

Хорошо, мы поняли, почему сократилось время ожидания в очереди. Но почему сокращение столь существенное (более чем в два раза)? Дело в том, что в данном примере обе одноканальных СМО работают почти на пределе своих возможностей. Стоит немного увеличить время обслуживания (т. е. уменьшить μ) и они перестанут справляться с потоком пассажиров, и очередь начнет неограниченно расти. А простои кассира в некотором смысле равносильны уменьшению его производительности μ. □

Пример 9.4. *На станции технического обслуживания автомобилей механики получают нужные им запчасти в отдел запчастей, где работают три клерка. Механики прибывают с интенсивностью* 40 *человек в час. Один клерк облуживает одного механика в среднем за три минуты.*

Владелец станции хочет определить нужно ли ему нанять еще одного клерка для работы за стойкой в отделе запчастей, если зарплата клерка в два раза меньше зарплаты механика.

Решение. Для $\lambda = 40$, $\mu = 60/3 = 20$, $\rho = \lambda/\mu = 40/20 = 2$ по формулам (9.22) и (9.23) вычислим среднее количество механиков, ждущих в очереди, когда число клерков n равно 3 и 4:

$$p_0(3) = \left(1 + \frac{\rho}{1!} + \frac{\rho^2}{2!} + \frac{\rho^3}{3!} + \frac{\rho^4}{3!(3-\rho)}\right)^{-1}$$

$$
\begin{aligned}
&= \left(1 + \frac{2}{1} + \frac{2^2}{2} + \frac{2^3}{6} + \frac{2^4}{6(3-2)}\right)^{-1} \\
&= (1 + 2 + 2 + 4/3 + 8/3)^{-1} = 1/9, \\
L_{\text{оч}}(3) &= \frac{\rho^{n+1} p_0}{n \cdot n!(1-\rho/n)^2} + \rho = \frac{2^4/9}{3 \cdot 6(1-2/3)^2} + 2 = 2\frac{8}{9}, \\
p_0(4) &= \left(1 + \frac{\rho}{1!} + \frac{\rho^2}{2!} + \frac{\rho^3}{3!} + \frac{\rho^4}{4!} + \frac{\rho^5}{4!(4-\rho)}\right)^{-1} \\
&= \left(1 + \frac{2}{1} + \frac{2^2}{2} + \frac{2^3}{6} + \frac{2^4}{24} + \frac{2^5}{24(4-2)}\right)^{-1} \\
&= (1 + 2 + 2 + 4/3 + 2/3 + 2/3)^{-1} = 3/23, \\
L_{\text{оч}}(4) &= \frac{\rho^{n+1} p_0}{n \cdot n!(1-\rho/n)^2} + \rho = \frac{2^5(3/23)}{4 \cdot 24(1-2/4)^2} + 2 = 2\frac{4}{23}.
\end{aligned}
$$

Поскольку $L_{\text{оч}}(3) - L_{\text{оч}}(4) = 2\frac{8}{9} - 2\frac{4}{23} = \frac{8 \cdot 23 - 4 \cdot 9}{9 \cdot 23} > 0.71 > 0.5$, то дополнительный клерк в отделе запчастей позволит сократить, которое механики проводят в отделе запчастей, боле чем на половину рабочего дня одного механика. Поскольку стоимость половины рабочего дня механика равна стоимости одного рабочего дня клерка, то мы можем рекомендовать хозяину станции нанять еще одного клерка. □

9.7. Упражнения

9.1. Кофе-автомат установлен в университетской столовой. В среднем за минуту к автомату подходят 3 студента. Каждому студенту в среднем требуется 15 секунд, чтобы обслужить себя. Ответьте на следующие вопросы:

а) в среднем какое число студентом можно увидеть у автомата;

б) в среднем сколько времени тратит студент, чтобы получить свою чашку кофе;

в) какой процент времени автомат простаивает;

г) какова вероятность того, что три или более студентов стоят у автомата.

9.2. Станция технического обслуживания автомобилей имеет всего одного механика, который специализируется на замене глушителей. В среднем за час приезжает два автомобиля для замены глушителя. Среднее время замены глушителя равно 20 минут. а) Каково среднее время нахождения автомобиля на станции. б) Сколько механиков должно работать на станции, если интенсивность потока прибывающих автомобилей увеличится вдвое, при условие, что среднее време нахождения автомобиля на станции не должно превышать одного часа.

Приложение A Элементы нелинейного анализа

A.1. Векторы и линейные простанства

Декартовым произведением множеств X и Y называется множество

$$X \times Y \stackrel{\text{def}}{=} \{(x, y) :\ x \in X,\ y \in Y\}$$

всевозможных пар элементов (x, y), где первый элемент x принадлежит X, а второй y принадлежит Y. Декартово произведение $X \times X$ обозначают через X^2. По индукции мы можем определить n-ю *степень множества* X:

$$X^n \stackrel{\text{def}}{=} X^{n-1} \times X = X \times X \times \cdots \times X.$$

Элементами множества X являются все упорядоченные наборы

$$(x_1, x_2, \ldots, x_n),$$

где $x_j \in X$, $j = 1, \ldots, n$.

Если $X = \mathbb{R}$ есть множество всех действительных чисел, то $\mathbb{R}^n$ есть множество действительных *векторов* (или *точек*) размерности n. В дальнейшем мы будем рассматривать n-мерные вектора как матрицы размера $n \times 1$, т. е. столбики, а чтобы представить вектор как строчку, будем использовать знак транспонирования:

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, \quad x^T = (x_1, x_2, \ldots, x_n).$$

Общепринято обозначать через e_i *i-й единичный вектор* $(0,\dots,1,\dots,0)^T$, который состоит из $n-1$-го нуля и одной единицы в позиции i. Для любого $x\in\mathbb{R}^n$ справедливо представление:

$$x = x_1\mathrm{e}_1 + x_2\mathrm{e}_2 + \cdots + x_n\mathrm{e}_n = \sum_{j=1}^{n} x_j\mathrm{e}_j\,.$$

Скалярное произведение векторов x и y из $\mathbb{R}^n$ определяется по правилу:

$$x^Ty = y^Tx = x_1y_1 + x_2y_2 + \cdots + x_ny_n = \sum_{j=1}^{n} x_jy_j.$$

Норма (или *длина*) вектора $x\in\mathbb{R}^n$ — это число $\|x\| \stackrel{\text{def}}{=} \sqrt{x^Tx}$. Расстояние между векторами $x,y\in\mathbb{R}^n$ определяется как длина вектора $x-y$, т. е. это число $\|x-y\|$.

Множество $\mathbb{R}^n$ с введенным на нем скалярным произведением называется *линейным* (или *евклидовым*) *пространством* $\mathbb{R}^n$.

Линейная, афинная и выпуклая оболочки множества векторов $X\subseteq\mathbb{R}^n$ соответственно задаются выражениями:

$$\begin{aligned}\mathrm{linhull}(X) \;&\stackrel{\text{def}}{=}\; \Big\{\sum_{i=1}^{k}\lambda_i x^i \,:\, k\ge 0\,; \\ &\qquad x^i\in X\,,\ \lambda_i\in\mathbb{R}\quad (i=1,\dots,k)\Big\}\,,\end{aligned} \tag{A.1}$$

$$\begin{aligned}\mathrm{affhull}(X) \;&\stackrel{\text{def}}{=}\; \Bigg\{\sum_{i=1}^{k}\lambda_i x^i \,:\, k\ge 1\,; \\ &\qquad x^i\in X\,,\ \lambda_i\in\mathbb{R}\quad (i=1,\dots,k)\,; \sum_{i=1}^{k}\lambda_i = 1\Bigg\}\,,\end{aligned} \tag{A.2}$$

$$\begin{aligned}\mathrm{affhull}(X) \;&\stackrel{\text{def}}{=}\; \Bigg\{\sum_{i=1}^{k}\lambda_i x^i \,:\, k\ge 1\,; \\ &\qquad x^i\in X\,,\ \lambda_i\in\mathbb{R}_+\quad (i=1,\dots,k)\,; \sum_{i=1}^{k}\lambda_i = 1\Bigg\}\,.\end{aligned} \tag{A.3}$$

Если $X = \mathrm{linhull}(X)$, то X называется *линейным подпространством*; X есть *афінное подпространство*, если $X = \mathrm{affhull}(X)$; если же $X = \mathrm{affhull}(X)$, то множество X *выпуклое*.

Базисом линейного подпространства $\mathcal{L} \subseteq \mathbb{R}^n$ называется минимальный набор векторов $\mathcal{B} = \{b^1, b^2, \dots, b^k\} \subseteq \mathcal{L}$, такой, что $\mathrm{linhull}(\mathcal{B}) = \mathcal{L}$. Из линейной алгебры известно, что каждый базис имеет одинаковое число векторов, которое называют *размером линейного подпространства* $\mathcal{L}$.

Отметим, что если $\mathcal{A} \subseteq \mathbb{R}^n$ есть афинное подпространство, то для любого $a \in \mathcal{A}$ множество $\mathcal{L} = \{x - a \,:\, x \in \mathcal{A}\}$ является линейным подпространством. Другими словами, афинное подпространство $\mathcal{A}$ можна определить как линейное подпространство $\mathcal{L}$, сдвинутое на вектор a: $\mathcal{A} = \mathcal{L} + a \stackrel{\mathrm{def}}{=} \{a + x \,:\, x \in \mathcal{L}\}$. *Размер* $\mathcal{A}$ определяется равным размеру $\mathcal{L}$. Афинное подпространство является линейным подпространством тогда и только тогда, когда оно содержит нулевой вектор. *Размер подмножества векторов* из $\mathbb{R}^n$ — это размер минимального афинного подпространства, которое содержит это подмножество.

Афинное подпространство векторного пространства $\mathbb{R}^n$ размера $n-1$ называется *гиперплоскостью.* Иначе, гиперплоскость $H(a, b)$ можно определить как множество точек $\{x \in \mathbb{R}^n \,:\, ax = b\}$, где $a \in \mathbb{R}^n$, $a \neq 0$, $b \in \mathbb{R}$. Гиперплоскость определяет два *полупространства* $H_{\leq}(a, b)$ и $H_{\geq}(a, b)$, которые соответственно определяются как множества точек $\{x \in \mathbb{R}^n \,:\, ax \leq b\}$ и $\{x \in \mathbb{R}^n \,:\, ax \geq b\}$.

А.2. Элементы топологии

Пусть $X \subseteq \mathbb{R}^n$. Элемент $x \in X$ называется *внутренней точкой* множество X, если существует такое $\epsilon > 0$, что $B(x, \epsilon) \subset X$. Здесь $B(x, \epsilon) \stackrel{\mathrm{def}}{=} \{y \in \mathbb{R}^n \,:\, \|y - x\| \leq \epsilon\}$ есть *шар* радиуса ϵ с центром x. Множество внутренних точек из X называется *внутренностью* множества X и обозначается $\mathrm{int}X$. Если $X = \mathrm{int}X$, то X —*открытое множество. Окрестностью* точки $x \in \mathrm{int}X$ называется любое открытое множество, которое содержит точку x.

Говорят, что $x \in \mathbb{R}^n$ есть *точка касания* множества $X \subset \mathbb{R}^n$, если $B(x, \epsilon) \cap X \neq \emptyset$ для любого $\epsilon > 0$. Множество всех точек касания множества X называется *замыканием множества* X и обозначается через $\mathrm{cl}X$. Множество X называется *замкнутым*, если $X = \mathrm{cl}X$. Множество $\mathrm{bd}X \stackrel{\mathrm{def}}{=} \mathrm{cl}X \setminus \mathrm{int}X$ называется *границей* множества X, а точки из $\mathrm{bd}X$ называются *граничными.*

Если размер множества $X \subset \mathbb{R}^n$ меньше n, то тогда $\mathrm{int}X = \emptyset$. Пусть $\mathcal{A}$ есть минимальное афинное подпространство, которому принадлежит множество X. *Относительная внутренность* $\mathrm{rint}(X)$ множества X есть множество точек $x \in X$, таких, что $B(x, \epsilon) \cap \mathcal{A} \subset X$ для некоторого $\epsilon > 0$.

Множество X называется *ограниченным*, если оно содержится в некотором шаре.

А.2.1. Компактные множества. Теорема Вейерштрасса

Бесконечную последовательность

$$x^1, x^2, \dots, x^k, \dots$$

векторов из $\mathbb{R}^n$ будем абозначать через $\{x^k\}_{k=1}^{\infty}$ или просто $\{x^k\}$. Говорят, что последовательность $\{x^k\}$ *сходится к точке* $x \in \mathbb{R}^n$ (или x есть *предел последовательности* $\{x^k\}$, пишут $x^k \to x$), если

$$\lim_{k\to\infty} \|x^k - x\| = 0.$$

Множество $X \subseteq \mathbb{R}^n$ называется *компактным*, если из любой последовательности $\{x^k\}_{k=1}^{\infty}$ элементов из X можно выбрать подпоследовательность $\{x^{k^i}\}_{i=1}^{\infty}$, которая сходится к некоторому элементу из X. Здесь $\{k^i\}$ есть неубывающая последовательность натуральных чисел. Известно, что в пространстве $\mathbb{R}^n$ множество X компактное тогда и только тогда, когда оно замкнуто и ограничено.

Следующая теорема является фундаментальной и касается существования оптимального решения в задачах оптимизации.

Теорема А.1 (Вейерштрасса). *Если f есть непрерывная функция на компактном множестве $X \in \mathbb{R}^n$, то задача*

$$\min_{x\in X} f(x)$$

имеет оптимальное решение $x^* \in X$ $(f(x^*) \le f(x)$ для всех $x \in X)$.

Следующее следствие из теоремы Вейерштрасса очень часто позволяет устанавить существование оптимального решения в задачах оптимизации без ограничений.

Следствие А.1. *Пусть f — непрерывная функция на $\mathbb{R}^n$, такая, что $f(x) \to \infty$, если $\|x\| \to \infty$. Тогда задача безусловной оптимизации*

$$\min_{x\in\mathbb{R}^n} f(x) \tag{A.4}$$

имеет оптимальное решение $x^* \in \mathbb{R}^n$.

A.3. Дифференцируемые функции

Часто бывает так, что функция f определена не на всем пространстве $\mathbb{R}^n$, а только на некотором падмножестве $X \subset \mathbb{R}^n$. В этом случае множество X определения функции f обозначают через $\mathrm{dom}(f)$ и называют *эффективной областью* функции f. Удобно считать, что $f(x) = \infty$ во всех точках $x \in \mathbb{R}^n \setminus \mathrm{dom}(f)$, а арифметические операции и операции сравнения для всех $q \in \mathbb{R}$ выполняются по следующим правилам:

$$\begin{array}{lll} q < \infty, & \max\{q, \infty\} = \infty, & \infty \le \infty, \\ q + \infty = \infty, & \infty + \infty = \infty, & \\ 0 \times \infty = 0, & t \times \infty = \infty & \text{для } t > 0 \end{array} .$$

Теперь $\mathrm{dom}(f) \stackrel{\mathrm{def}}{=} \{x \in \mathbb{R}^n \, : \, f(x) < \infty\}$.

Пусть $f : \mathbb{R}^n \to \mathbb{R}$. Если для $x \in \mathrm{dom}(f)$ и $p \in \mathbb{R}^n$ существует предел

$$\lim_{\epsilon \to 0} \frac{f(x + \epsilon p) - f(x)}{\epsilon},$$

то он называется *производной по направлению* p функции f в точке x и обозначается через $\frac{\partial f}{\partial p}(x)$. Отметим также, что если рассматривать $\frac{\partial f}{\partial p}(x)$ как функцию от x, то для $q \in \mathbb{R}^n$ можно определить

$$\frac{\partial^2 f}{\partial q \, \partial p}(x) \stackrel{\mathrm{def}}{=} \frac{\partial}{\partial q} \left(\frac{\partial f}{\partial p} \right)(x).$$

Для $i = 1, \dots, n$ величина $\frac{\partial f}{\partial \mathrm{e}_i}(x)$ обозначается через $\frac{\partial f}{\partial x_i}(x)$ и называется *частной производной* по координате x_i. Если в точке x и в некоторой ее окрестности существуют частные производные $\frac{\partial f}{\partial x_i}(x)$ для $i = 1, \dots, n$, то говорят, что функция f *дифференцируема* в точке x. Вектор, составленный из всех n частных производных называется *градиентом* функции f в точке x. Мы будем обозначать его через

$$\nabla f(x) \stackrel{\mathrm{def}}{=} \left(\frac{\partial f}{\partial x_1}(x), \dots, \frac{\partial f}{\partial x_n}(x) \right)^T.$$

По аналогии с одномерным случаем, можно определить производные высших порядков как производные от производных предшествующих порядков. При этом, число частных производных следующего порядка в n раз больше числа производных предшествующего порядка. В оптимизации, как правило, не используют производные порядка выше второго.

Можно определить n^2 *вторых* частных производных:

$$\frac{\partial}{\partial x_i}\left(\frac{\partial f}{\partial x_j}\right)(x), \quad i = 1, \dots, n;\ j = 1, \dots, n.$$

Эти величины обычно записывают так:

$$\frac{\partial^2 f}{\partial x_i \partial x_j}(x), \quad i \neq j; \quad \frac{\partial^2 f}{\partial^2 x_i}(x), \quad i = j.$$

Если частные производные $\partial f(x)/\partial x_i$, $\partial f(x)/\partial x_j$ и $\partial^2 f(x)/\partial x_i \partial x_j$ существуют и непрерывны, то существует и $\frac{\partial^2 f}{\partial x_j \partial x_i}(x)$, причем

$$\frac{\partial^2 f}{\partial x_i \partial x_j}(x) = \frac{\partial^2 f}{\partial x_j \partial x_i}(x).$$

В этом случае все n^2 частных производных второго порядка принято сводить в квадратную симметричную *матрицу вторых производных*, которую также называют *матрицей Гессе*. В дальнейшем эту матрицу будем обозначать через

$$\nabla^2 f(x) \stackrel{\text{def}}{=} \begin{bmatrix} \dfrac{\partial^2 f}{\partial^2 x_1}(x) & \dots & \dfrac{\partial^2 f}{\partial x_1 \partial x_n}(x) \\ \dots\dots & \dots & \dots\dots \\ \dfrac{\partial^2 f}{\partial x_n \partial x_1}(x) & \dots & \dfrac{\partial^2 f}{\partial^2 x_n}(x) \end{bmatrix}.$$

В оптимизации находят применения многие результаты класического анализа. Но наиболее часто при аппроксимации функции в окрестности некоторой точки применяется теорема Тейлора. В оптимизационных алгоритмах, как правило, редко используется более трех членов ряда Тейлора. Пусть $x \in \mathbb{R}^n$ — некоторая точка, а $p \in \mathbb{R}^n$ — вектор, который задает некоторое направление. Тогда в окрестности точки $h = 0$ имеет место равенство

$$f(x + hp) = f(x) + h(\nabla f(x))^T p + \frac{1}{2} h^2 p^T \nabla^2 f(x)\, p + O(h^3).$$

Отметим, что скорость изменения функции f при движении из точки x вдоль направления p задается величиной $(\nabla f(x))^T p$, которую называют *первой производной по направлению* p. Аналогично, число $p^T \nabla^2 f(x)\, p$ называется *второй производной по направлению* p. Ее еще называют *кривизной* f вдоль направления p. Если $p^T \nabla^2 f(x)\, p > 0$ (< 0), то говорят, что p — направление *положительной* (*отрицательной*) кривизны.

А.4. Необходимые условия локального минимума

Точка $x^0 \in \mathbb{R}^n$ называется *локальным минимумом функции* $f : \mathbb{R}^n \to \mathbb{R}$ *на множестве* X, если существует окрестность $B(x^0, r)$ точки x^0, что $f(x^0) \leq f(x)$ для всех $x \in B(x^0, r) \cap X$. Точка x^0 есть *глобальный минимум функции* $f(x)$ *на* X, если $f(x^0) \leq f(x)$ для всех $x \in X$. Если $X = \mathbb{R}^n$, то мы будзем говорить про *локальный и глобальный минимумы функции* f.

Если $f(x)$ — дважды непрерывно дифференцируемая функция, то, используя формулу Тейлора в окрестности точки $x^* \in \mathbb{R}^n$

$$f(x^* + hp) = f(x^*) + hp^T \nabla f(x^*) + \frac{1}{2} h^2 p^T \, \nabla^2 f(x^* + \theta hp) \, p,$$

где $h \geq 0$, $0 \leq \theta \leq 1$, а $p \in \mathbb{R}^n$, нетрудно получить:

1. **Необходимые условия локальнога минимума**:

   (У1) x^* —*стационарная точка*, т.е. $\nabla f(x^*) = 0$;

   (У2) матрица $\nabla^2 f(x^*)$ неотрицательно определена.

2. **Достаточные условия локального минимума**:

   (Д1) $\nabla f(x^*) = 0$;

   (Д2) матрица $\nabla^2 f(x^*)$ положительно определена.

А.5. Выпуклые множества

Как мы уже отмечали, множество $X \subseteq \mathbb{R}^n$ называется выпуклым, если $X = \text{affhull}(X)$. Это определение эквивалентно следующему более простому определению. Множество $X \subseteq \mathbb{R}^n$ называется *выпуклым*, если вместе с любыми своими двумя точками $x, y \in X$ оно содержит и отрезок

$$[x, y] \stackrel{\text{def}}{=} \{z(\lambda) = (1 - \lambda)x + \lambda y \; : \; 0 \leq \lambda \leq 1\},$$

соединяющий эти точки. Простейшими примерами выпуклых множеств являются:

- *линейное пространство* $\mathbb{R}^n$ и *положительный ортант*

$$\mathbb{R}^n_+ \stackrel{\text{def}}{=} \{x \in \mathbb{R}^n \; : \; x_i \geq 0, \; i = 1, \ldots, n\};$$

- евклидов шар $B(x^0, r)$, $x^0 \in \mathbb{R}^n$, $r \in \mathbb{R}_{++} \stackrel{\text{def}}{=} \{\alpha \in \mathbb{R} : \alpha > 0\}$;
- гиперплоскость $H(a, b)$ и полупространства $H^{\leq}(a, b)$, $H^{\geq}(a, b)$.

Нетрудно убедиться, что пересечение $X \cap Y$ двух выпуклых множеств $X, Y \subseteq \mathbb{R}^n$ — также выпуклое множество. Отсюда следует, что и *полиэдр*, который определяется как множество решений системы линейных неравенств $Ax \leq b$, — также выпуклое множество, поскольку оно является пересечение конечного числа полупространств. В частности, n-мерный *симплекс*

$$\Sigma_n \stackrel{\text{def}}{=} \{x \in \mathbb{R}^n_+ : \sum_{j=1}^{n} x_j = 1\}$$

также является выпуклым множеством.

A.5.1. Выпуклые конусы

Множество $C \in \mathbb{R}^n$ называется *выпуклым конусом*, если оно замкнуто относительно умножения на положительные скаляры (если $x \in C$, то $tx \in C$ для всех $t > 0$) и относительно сложения (из $x, y \in C$ следует $x + y \in C$). Конус C определяет *упорядочение* на $\mathbb{R}^n$: $x \geq_C y$ означает, что $x - y \in C$. Мы также будем использовать обозначение $x >_C y$, если $x - y \in \text{int}C$. Как обычно, мы пишем $x \leq_C y$ ($x <_C y$), если $y \geq_C x$ ($y >_C x$). Введенное упорядочение $\geq_C$ является *частичным порядком*, если конус C *острый*, т. е. $C \cap -C = \{0\}$.

Конусом, порожденным векторами $a^1, \ldots, a^m \in \mathbb{R}^n$, называется множество

$$\text{cone}(a^1, \ldots, a^m) \stackrel{\text{def}}{=} \{x \in \mathbb{R}^n \,:\, x = \sum_{i=1}^{m} y_i a^i \,,\, y \in \mathbb{R}^m_+\}$$

Такие конусы еще называют *конечнопорожденными*. Конус $C = \{x \in \mathbb{R}^n \,:\, Ax \geq 0\}$, где $A \in M_{m,n}(\mathbb{R})$ называется *полиэдральным*. Известно, что выпуклый конус C является полиэдральным тогда и только тогда, когда он конечнопорожденный. Другими словами, понятия «полиэдральный конус» и «конечнопорожденный конус»— эквивалентны.

Двойственный конус для конуса $C \subseteq \mathbb{R}^n$ обозначается через C^D и определяется следующим образом

$$C^D \stackrel{\text{def}}{=} \{y \in \mathbb{R}^n : y^T x \geq 0 \text{ для всех } x \in C\}.$$

Отметим, что $(\mathbb{R}^n_+)^D = \mathbb{R}^n_+$ i $(SM^n_+)^D = SM^n_+$.

А.5.2. Теорема об отделении выпуклых множеств

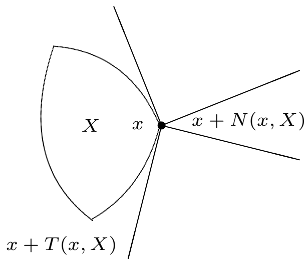


Рис. А.1. Касательный и нормальный конусы

Пусть $X \subset \mathbb{R}^n$ есть выпуклое множество и $x \in X$. Обозначим через

$$S(x,X) \stackrel{\text{def}}{=} \cup_{t>0} \frac{1}{t}(X-x)$$

конус, порожденный множеством $X-x$, и через $T(x,X) \stackrel{\text{def}}{=} \mathrm{cl}(S(x,X))$ — его замыкание. Множество $T(x,X)$ называется *касательным конусом* к X в точке x. Нетрудно убедиться, что $S(x,X)$ и $T(x,X)$ — выпуклые конусы. Отметим также, что

$$X \subset x + S(x,X) \subset x + T(x,X).$$

Конус $N(x,X) \stackrel{\text{def}}{=} -T(x,X)^D$ называется *нормальным конусом* к X в точке x (см. рис. А.1).

Проекцией точки x на множество X называется такая точка $y \in \mathrm{cl}X$, что

$$\|x-y\| \le \|x-z\| \quad \text{для всех } z \in \mathrm{cl}X.$$

Для любой точки $x \in \mathbb{R}^n$ существует единственная ее проекция $\mathrm{pr}(x,X)$ на выпуклое множество X. Этот факт следует из строгой выпуклости функции $f(y) \stackrel{\text{def}}{=} \|y-x\|$, определенной на множестве X. Как мы уже отмечали выше, если X есть линейное подпространство, то $\mathrm{pr}(x,X) = P_X x$.

Теорема А.2. *Пусть $X \subset \mathbb{R}^n$ есть выпуклое множество и $x \in X$. Тогда*

$$\mathrm{pr}^{-1}(x,X) \stackrel{\text{def}}{=} \{y \in \mathbb{R}^n : \mathrm{pr}(y,X) = x\} = x + N(x,X).$$

Следующий результат, известный как теорема об *отделении выпуклых множеств*, можно получить как следствие из теоремы А.2.

Теорема А.3. *Пусть X, Y — непустые выпуклые множества в $\mathbb{R}^n$. Если $X \cap Y = \emptyset$, то существует отделяющая их гиперплоскосць $H(a,b)$, такая, что*

$$a^T x \le b < a^T y \quad \textit{для всех } x \in X,\ y \in Y.$$

Если же только $\mathrm{int}X \cap \mathrm{int}Y = \emptyset$, *то существует гиперплоскость $H(a,b)$, такая, что*

$$a^T x \le b \le a^T y \quad \textit{для всех } x \in X,\ y \in Y.$$

А.6. Лемма Фаркаша

Чтобы понять причину несовместности системы линейных неравенств, рассмотрим простой пример:

$$\begin{aligned} 2x_1 + 5x_2 + x_3 &\le 5, \\ x_1 + 2x_2 &\ge 3, \\ x_2 &\ge 0, \\ x_3 &\ge 0. \end{aligned}$$

Сложив первое неравенство со вторым, умноженным на -2, третьим и четвертым, умноженными на -1, получим ложное неравенство $0 \le -1$. Отсюда мы можем сделать вывод, что рассматриваемая система неравенств несовместна. Как ни странно, но система линейных неравенств несовместна тогда и только тогда, когда из нее можно вывести ложное неравенство $0 \le -1$. Дадим более точную формулировку данного критерия, известного как *лемма Фаркаша*.

Лемма А.1. *Система неравенств $Ax \le b$ не имеет решения тогда и только тогда, когда существует такой вектор $y \ge 0$, что $y^T A = 0$ и $y^T b < 0$.*

А.7. Выпуклые функции

Функция $f : X \to \mathbb{R}$, определенная на выпуклом множестве X, называется *выпуклой*, если для всех $x, y \in X$ и любого $\lambda \in [0, 1]$ выполняется неравенство

$$f((1-\lambda)x + \lambda y) \le (1-\lambda)f(x) + \lambda f(y). \tag{A.5}$$

Функция f называется *вогнутой*, если функция $-f$ выпукла. В дальнейшем мы будем в основном рассматривать свойства выпуклых функций. Заменяя в этих свойствах знаки неравенств на противоположные и минимум на максимум, вы получите соответствующие свойства вогнутых функций.

Для выпуклой функции $f : X \to \mathbb{R}$, определенной на выпуклом множестве X, и любого $\alpha \in \mathbb{R}$, если множество

$$\{x \in X : \ f(x) \le \alpha\}$$

не пустое, то оно выпукло. В частности, выпукло множество

$$\arg\min_{x \in X} f(x) \stackrel{\text{def}}{=} \{x \in X : \ f(x) \le \min\{f(x) : \ x \in X\}\}$$

всех минимумов функции f.

Важно отметить, что *локальный* минимум выпуклой функции на выпуклом множестве является *глобальным.* Действительно, пусть $x^1, x^2 \in X$ есть локальные минимумы функции f на выпуклом множестве $X \subseteq \mathbb{R}^n$, причем $f(x^1) > f(x^2)$. Пусть $f(x^1) \le f(x)$ для всех $x \in B(x_1, \epsilon) \cap X$ для некоторого $\epsilon > 0$. Так как $x^2 \notin B(x_1, \epsilon)$, то $\lambda = \epsilon / \|x^2 - x^1\| < 1$ и в силу выпуклости функции f выполняется неравенство

$$f((1-\lambda)x^1 + \lambda x^2) \le (1-\lambda) * f(x^1) + \lambda f(x^2) < f(x^1).$$

В силу выбора λ, точка $(1-\lambda)x^1 + \lambda x^2$ принадлежит шару $B(x_1, \epsilon)$ и, следовательно, $x^1 \in X$ не является точкой локального минимума.

Если (A.5) всегда выполняется как строгое неравенство, то функция f называется *строго выпуклой.* Нетрудно убедиться, что строго выпуклая функция f на любом выпуклом множестве X имеет единственный минимум, т.е. существует точка $x^* \in X$, что $f(x^*) < f(x)$ для всех $x \in X$.

Используя формулу Тейлора, можно получить следующие критерии выпуклости гладкой функции.

Теорема А.4. *а) Непрерывно дифференцируемая функция* $f : \mathbb{R}^n \to \mathbb{R}$ *выпукла тогда и только тогда, когда*

$$f(y) - f(x) \ge (\nabla f(x))^T (y - x) \quad \textit{для всех} \quad x, y \in \mathbb{R}^n, \tag{A.6}$$

где

$$\nabla f(x) \stackrel{\text{def}}{=} \left(\frac{\partial f}{\partial x_1}(x), \dots, \frac{\partial f}{\partial x_n}(x) \right)^T$$

есть градиент*функции* f *в точке* x.

б) Дважды непрерывно дифференцируемая функция $f : \mathbb{R}^n \to \mathbb{R}$ *выпукла (строго выпукла) тогда и только тогда, когда в любой точке* $x \in \mathbb{R}^n$ *матрица вторых производных (Гессиан)*

$$\nabla^2 f(x) \stackrel{\text{def}}{=} \begin{bmatrix} \frac{\partial^2 f}{\partial^2 x_1}(x) & \dots & \frac{\partial^2 f}{\partial x_1 \partial x_n}(x) \\ \dots & \dots & \dots \\ \frac{\partial^2 f}{\partial x_n \partial x_1}(x) & \dots & \frac{\partial^2 f}{\partial^2 x_n}(x) \end{bmatrix}$$

неотрицательно определена (положительно определена).

В частности, из теоремы А.4 следует, что квадратичная функция $f(x) = c^T x + x^T Q x$ выпукла на $\mathbb{R}^n$ тогда и только тогда, когда матрица Q неотрицательно определена.

А.7.1. Как доказать выпуклость функции

Выпуклость или вогнутость конкретной функции можно доказать разными способами.

1. По определению, проверив выполнимость неравенства (А.5). Функция $f(x) = \max\{x_1, \ldots, x_n\}$ является выпуклой на $\mathbb{R}^n$, поскольку для любых $x, y \in \mathbb{R}^n$ и всех $\lambda \in [0,1]$ справедливо

$$\begin{aligned} f((1-\lambda)x + \lambda y) &= \max_{1\le i\le n}((1-\lambda)x_i + \lambda y_i) \\ &\le (1-\lambda)\max_{1\le i\le n} x_i + \lambda \max_{1\le i\le n} y_i \\ &= (1-\lambda)f(x) + \lambda f(y). \end{aligned}$$

2. Показать, что Гессиан является неотрицательно определенной матрицей. Функция $f(x,y) = x^2/y$ выпукла на $\mathbb{R}\times\mathbb{R}_{++}$, потому что матрица

$$\nabla^2 f(x,y) = \frac{2}{y^3}\begin{bmatrix} y^2 & -xy \\ -xy & x^2 \end{bmatrix} = \frac{2}{y^3}\begin{bmatrix} y \\ -x \end{bmatrix}\begin{bmatrix} y & -x \end{bmatrix}$$

неотрицательно определена[13].

Чуть труднее доказать, что функция $f(x) = \left(\prod_{i=1}^n x_i\right)^{1/n}$, значения которой равны среднему геометрическому ее аргументов, является вогнутой на $\mathbb{R}_{++}$. Компоненты ее Гессиана $\nabla^2 f(x)$ вычисляются по правилу:

$$\begin{aligned} \frac{\partial^2 f}{\partial^2 x_k}(x) &= -(n-1)\frac{\left(\prod_{i=1}^n x_i\right)^{1/n}}{n^2 x_k^2}, \quad k = 1, \ldots, n, \\ \frac{\partial^2 f}{\partial x_k \partial x_l}(x) &= \frac{\left(\prod_{i=1}^n x_i\right)^{1/n}}{n^2 x_k x_l}, \quad k, l = 1, \ldots, n,\ k \ne l. \end{aligned}$$

Поскольку для произвольного вектора $y \in \mathbb{R}^n$ выполняется неравенство

$$y^T \nabla^2 f(x) y = -\frac{\left(\prod_{i=1}^n x_i\right)^{1/n}}{n^2}\left(n\sum_{i=1}^n (y_i/x_i)^2 - \left(\sum_{i=1}^n y_i/x_i\right)^2\right) \le 0,$$

то матрица $\nabla^2 f(x)$ неположительно определена. Здесь мы использовали неравенство Коши-Шварца $|u^T v| \le \|u\|\,\|v\|$ с $u_i = 1$ и $v_i = y_i/x_i$, $i = 1, \ldots, n$.

[13] Матрица A размера $n\times n$ неотрицательно определена тогда и только тогда, когда существует $m \times n$ матрицы B, что $A = B^T B$.

3. Доказать, что ограничение многомерной функции на произвольный отрезок является одномерной выпуклой функцией.

А.7.2. Преобразования, сохраняющие выпуклость функций

Еще один общий способ доказать выпуклость (вогнутость) некоторой функции состоит в том, чтобы показать, что рассматриваемая функция получается из одной или нескольких известных выпуклых (вогнутых) функций применением преобразований, которые сохраняют выпуклость функций.

1. Неотрицательная взвешенная сумма. Если f — выпуклая функция и $\alpha \geq 0$, то, очевидно, и функция αf — также выпуклая функция. Далее, если $f, g : X \to \mathbb{R}$ есть выпуклые функции, определенные на выпуклом множестве X, то их сумма $h(x) = f(x) + g(x)$ также будет выпуклой на X. Комбинируя эти две операции, мы получим, что взвешенная сумма $f = \sum_{i=1}^{m} f_i$ выпуклых функций $f_1, \dots, f_m$ с неотрицательными весами $w_1, \dots, w_n$ также является выпуклой функцией.

Эти свойства распространяются на бесконечные суммы и интегралы. Например, если $f(x, y)$ — выпуклая функция аргумента $x \in X$ для всех $y \in Y$ и $w(y) \geq 0$ для всех $y \in Y$, то функция g, определенная по правилу

$$g(x) = \int_Y w(y) f(x, y) dy$$

является выпуклой на X, при условии, что интеграл существует.

2. Аффинные преобразования. Пусть $f : \mathbb{R}^n \to \mathbb{R}$, A — $m \times n$-матрица и $b \in \mathbb{R}^n$. Определим функцию $g : \mathbb{R}^m \to \mathbb{R}$ по правилу $g(x) = f(Ax + b)$. Тогда, если функция f выпуклая (вогнутая), то и g — также выпуклая (вогнутая) функция.

3. Поточечный максимум. Если f и g есть выпуклые функции на X, то выпуклой функцией является и их поточечный максимум $h(x) = \max\{f(x), g(x)\}$. Действительно, для $x, y \in X$ и $\lambda \in [0, 1]$ имеем

$$\begin{aligned}
h((1-\lambda)x + \lambda y) &= \max\{f((1-\lambda)x + \lambda y), g((1-\lambda)x + \lambda y)\} \\
&\leq \max\{(1-\lambda)f(x) + \lambda f(y), (1-\lambda)g(x) + \lambda g(y)\} \\
&\leq (1-\lambda)\max\{f(x), g(x)\} + \lambda \max\{f(y), g(y)\} \\
&= (1-\lambda)h(x) + \lambda h(y),
\end{aligned}$$

что доказывает выпуклость функции h.

Если $f_1, \ldots, f_m$ — выпуклые на X функции, то и их поточный максимум

$$f(x) = \max\{f_1(x), \ldots, f_m(x)\}$$

является выпуклой на X функцией.

В частности, кусочно-линейная функция $f(x) = \max_{1 \le i \le m}(c^i)^T x$, где $c^i \in \mathbb{R}^n$ для $i = 1, \ldots, m$, является выпуклой на $\mathbb{R}^n$. Заметим, что в общем случае такие выпуклые функции не являются дифференцируемыми.

4. Композиция. *Композиция* функций $f : \mathbb{R}^n \to \mathbb{R}$ и $g : \mathbb{R} \to \mathbb{R}$ есть функция $h = g \circ f : \mathbb{R}^n \to \mathbb{R}$, определяемая по правилу: $h(x) = g(f(x))$. Справедливы следующие утверждения:

- h выпуклая, если f выпуклая, а g неубывающая и выпуклая;
- h выпуклая, если f вогнутая, а g невозрастающая выпуклая;
- h вогнутая, если f вогнутая, а g неубывающая вогнутая;
- h вогнутая, если f выпуклая, а g невозрастающая вогнутая.

Если $n = 1$ и обе функции f и g являются дважды дифференцируемыми, то сформулированные выше утверждения следуют из равенства

$$h''(x) = g''(f(x))f'(x)^2 + g'(f(x))f''(x).$$

А.7.3. Субградиенты и субдифференциал

Для дифференцируемой выпуклой функции f градиент $\nabla f(x^0)$ в точке x^0 удовлетворяет фундаментальному неравенству (А.6). Для не всюду дифференцируемой выпуклой функции можно ввести понятие субградиента, которое обобщает понятие градиента.

Субградиентом функции $f : \mathbb{R}^n \to \mathbb{R}$ в точке $x^0 \in \mathbb{R}^n$ называется вектор $\gamma \in \mathbb{R}^n$, удовлетворяющий условию

$$f(x) - f(x^0) \ge \gamma^T (x - x^0_x). \tag{А.7}$$

В пространстве $\mathbb{R}^{n+1}$ гиперплоскость, определяемая равенством

$$y - \gamma^T (x - x^0) = f(x^0),$$

является *опорной* для надграфика $G(f) \stackrel{\text{def}}{=} \{(x, y) \in X \times \mathbb{R} : y \ge f(x)\}$ функции f в точке $(x^0, f(x^0))$, т. е. эта гиперплоскость касается надграфика в точке $(x^0, f(x^0))$ и весь надграфик лежит по одну сторону от гиперплоскости.

Фундаментальным результатом является следующая теорема.

Теорема А.5. *Любая выпуклая функция $f : X \to \mathbb{R}$, определенная на выпуклом множестве $X \subseteq \mathbb{R}^n$, имеет субградиент в любой точке $x^0 \in \mathrm{rint}(X)$.*

Субдифференциалом функции f в точке x^0 — обозначается $\partial f(x^0)$ — называется множество всех субградиентов функции f в точке x^0.

Если функция f дифференцируема в точке x^0, то $\partial f(x^0) = \{\nabla f(x^0)\}$.

Теорема А.6. *Пусть $f : X \to \mathbb{R}$ есть выпуклая функция, определенная на выпуклом множестве $X \subseteq \mathbb{R}^n$. Точка $x^0 \in X$ есть точка минимума функции f тогда и только тогда, когда $0 \in \partial f(x^0)$.*

Доказательство. Действительно, $0 \in \partial f(x^0)$ тогда и только тогда, когда

$$f(x) \geq f(x^0) + 0^T(x - x^0) \quad \text{для всех } x \in X;$$

а это неравенство в свою очередь выполняется в том и только том случае, если x^0 есть точка минимума функции f на множестве X. □

А.8. Квазивыпуклые функции

Для выпуклой функции $f : X \to \mathbb{R}$, определенной на выпуклом множестве X, и любого $\alpha \in \mathbb{R}$, если множество

$$S^f_\alpha \stackrel{\text{def}}{=} \{x \in X :\ f(x) \leq \alpha\}$$

не пустое, то оно выпукло. В частности, выпукло множество

$$\arg\min_{x \in X} f(x) \stackrel{\text{def}}{=} \{x \in X :\ f(x) \leq \min\{f(x) :\ x \in X\}\}$$

всех минимумов функции f. Класс функций, которые обладают этим важным свойством существенно шире класса выпуклых функций.

Функция f называется *квазивыпуклой* на выпуклом множестве $X \subseteq \mathbb{R}^n$, если для любого $\alpha \in \mathbb{R}$ множество S^f_α или выпуклое или пустое. Функция f называется *квазивогнутой*, если $-f$ квазивыпуклая функция.

Пример квазивыпуклой функции на $\mathbb{R}$ приведен на рис. А.2, а. Для любого α, если $S^f_\alpha \neq \emptyset$, то S^f_α является интервалом. В частности, $S^f_{\alpha_1} = [a, b]$, а $S^f_{\alpha_2} = (-\infty, c]$.

Непрерывная функция на $f : \mathbb{R} \to \mathbb{R}$ является квазивыпуклой тогда и только тогда, когда выполняется одно из следующих условий:

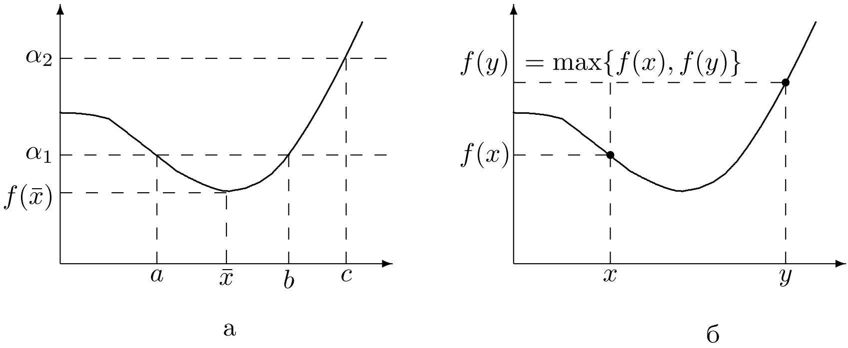


Рис. А.2. Квазивыпуклая функция на $\mathbb{R}$

- f монотонная (неубывающая или невозрастающая) функция;
- для любой точки $\bar{x}$ своего глобального минимума, функции f невозрастает на интервале $(-\infty, \bar{x}]$ и неубывает на интервале $[\bar{x}, \infty)$.

Если $f : X \to \mathbb{R}_+$ и $g : X \to \mathbb{R}_{++}$ соответственно выпуклая и вогнутая функции, определенные на выпуклом множестве $X \subseteq \mathbb{R}^n$, то их отношение $f(x) = g(x)/h(x)$ является квазивыпуклой функцией. Действительно, для произвольного $\alpha \in \mathbb{R}$ множество

$$S_\alpha^f = \{x \in X : g(x)/h(x) \le \alpha\} = \{x \in X : g(x) - \alpha h(x) \le 0\}$$

выпукло. Этот пример еще раз доказывает, что класс квазивыпуклых функций существенно шире класса выпуклых функций,

А.8.1. Критерии квазивыпуклости функций

Теорема А.7. *Функция $f : X \to \mathbb{R}$, определенная на выпуклом множестве X, квазивыпукла, тогда и только тогда, когда для всех $x, y \in X$ и $0 \le \lambda \le 0$*

$$f((1-\lambda)x + \lambda y) \le \max\{f(x), f(y)\}. \tag{А.8}$$

Неравенство (А.8) означает, что значение функции в любой точке отрезка не превышает максимального значение функции на концах этого отрезка (см. рис. А.2, б).

Для дифференцируемых функций справедливы следующие критерии.

Теорема А.8. *Непрерывно дифференцируемая функция $f : X \to \mathbb{R}$, определенная на выпуклом множестве X, квазивыкла, тогда и только тогда, когда для всех $x, y \in X$*

$$f(y) \le f(x) \Rightarrow (\nabla f(x))^T (y - x) \le 0. \tag{A.9}$$

Если $\nabla f(x) \neq 0$, то неравенство (A.9) означает, что вектор $\nabla f(x)$ является нормалью к касательной гиперплоскости в точке x к множеству $\{y : f(y) \le f(x)\}$. Понятно, что неравенство (A.9) верно и для выпуклых функций. Но между выпуклыми и квазивыпуклыми функциями имеется существенное различие: если f — выпуклая функция и $\nabla f(x) = 0$, то x есть точка глобального минимума функции f; но для квазивыпуклой функции f из $\nabla f(x) = 0$ не следует, что x есть ее точка глобального минимума.

Теорема А.9. *Дважды непрерывно дифференцируемая функция $f : X \to \mathbb{R}$, определенная на выпуклом множестве X, квазивыкла, тогда и только тогда, когда для всех $x \in X$ и всех $y \in \mathbb{R}^n$*

$$y^T \nabla f(x) = 0 \Rightarrow y^T \nabla^2 f(x) y \ge 0. \tag{A.10}$$

Например, функция $f(x) = x_1 \cdot x_2$ является квазивогнутой (но не вогнутой) на открытом выпуклом множестве $\mathbb{R}^2_{++}$ и квазивыпуклой (но не выпуклой) на открытом выпуклом множестве $\mathbb{R}^2_{--}$, где $\mathbb{R}_{--} \stackrel{\text{def}}{=} \{x \in \mathbb{R} :\ x < 0\}$. Действительно, поскольку для $x \in \mathbb{R}^2_{++}$

$$\nabla f(x) = \begin{pmatrix} x_2 \\ x_1 \end{pmatrix}, \quad \nabla^2 f(x) = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix},$$

то для любого $y \in \mathbb{R}^2$, такого, что $y^T \nabla f(x) = y_1 x_2 + y_2 x_1 = 0$, имеем $y_1 = -(x_1/x_2) y_2$ и

$$y^T \nabla^2 f(x) y = (y_1, y_2) \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = 2 y_1 y_2 = -(x_1/x_2) y_2^2 < 0.$$

Поэтому $f(x) = x_1 \cdot x_2$ является квазивогнутой на $\mathbb{R}^2_{++}$.

А.8.2. Преобразования, сохраняющие квазивыпуклость функций

1. Неотрицательный взвешенный поточный максимум. Для неотрицательных весовых множителей $w_1, \ldots, w_m \ge 0$ и квазивыпуклых функций $f_1, \ldots, f_m$ функция f с

$$f(x) = \max\{f_1(x), \ldots, f_m(x)\}$$

является квазивыпуклой.

Верен и более общий результат. Пусть Y — произвольное множество, X — выпуклое множество из $\mathbb{R}^n$, $w(y) \geq 0$ для всех $y \in Y$, а $g(x, y)$ есть выпуклая на X функция аргумента x при всех фиксированных значениях $y \in Y$. Тогда функция

$$f(x) = \sup_{y \in Y} (w(y)g(x, y))$$

является квазивыпуклой на X. Этот факт проверяется просто: для заданного $\alpha \in \mathbb{R}$ множество S_α^f выпукло, поскольку оно совпадает с пересечением выпуклых множеств $S_\alpha^{f_y}$ для всех $y \in Y$, где $f_y(x) = w(y)g(x, y)$.

2. Композиция. Если $f : \mathbb{R}^n$ — квазивыпуклая функция, а $g : \mathbb{R} \to \mathbb{R}$ — неубывающая функция, то и композиция $g \circ f$ является квазивыпуклой функцией.

Композиция $f((Ax + b)/(c^T x + d))$ квазивыпуклой функции f, определенной на выпуклом множестве $X \subseteq \mathbb{R}^n$, и дробно-линейной функции $(Ax + b)/(c^T x + d)$ является квазивыпуклой функцией на множестве

$$\{x \in \mathbb{R}^n : | : c^T x + d > 0,\ (Ax + b)/(c^T x + d) \in X\}.$$

3. Минимизация. Если функция g является квазивыпуклой на $X \times Y$, где $X \subseteq \mathbb{R}^n$ и $Y \subseteq \mathbb{R}^m$ — выпуклые множества, то и функция

$$f(x) = \inf_{y \in Y} g(x, y)$$

является квазивыпуклой.

Приложение В
Элементы теории вероятностностей

В этом приложении в сжатой форме представлены те понятия теории вероятностей, которые необходимы для понимания материала данной книги. Предполагается, что вы уже прослушали начальный курс теории вероятностей. Это приложение поможет вам систематизировать свои знания.

В.1. Вероятностные пространства

Вероятностным пространством называется тройка $(\Omega, \mathcal{A}, \mathbb{P})$, где

- Ω — это некоторое множество, которое называется *пространством элементарных событий* или *выборочным пространством*;
- $\mathcal{A}$ — семейство подмножеств множества Ω, которое является *сигма-алгеброй*, т. е. семейство $\mathcal{A}$ должно обладать следующими свойствами:
  - (а) если $A \in \mathcal{A}$, то и $\bar{A} \stackrel{\mathrm{def}}{=} \Omega \setminus A \in \mathcal{A}$;
  - (б) если каждое из счетного числа множеств $A_1, A_2, \ldots$ принадлежит $\mathcal{A}$, то и их объединение $\cup_{i=1}^{\infty} A_i$ также принадлежит $\mathcal{A}$;
- $\mathbb{P} : \mathcal{A} \to [0, 1]$ — *вероятностная мера*, которая должна удовлетворять следующим условиям:
  - (i) для любого счетного семейства попарно непересекающихся подмножеств $A_1, A_2, \ldots$ $(A_i \cap A_j = \emptyset$, если $i \neq j)$ справедливо

равенство

$$\mathbb{P}\left(\cup_{i=1}^{\infty} A_i\right) = \sum_{i=1}^{\infty} \mathbb{P}(A_i);$$

(ii) $\mathbb{P}(\Omega) = 1$.

Из закона Де Моргана ($A \cap B = \overline{\bar{A} \cup \bar{B}}$ для любых $A, B \in \Omega$) следует, что в (б) знак объединения можно заменить на знак пересечения и получить условие

(в) если каждое из счетного числа множеств $A_1, A_2, \ldots$ принадлежит $\mathcal{A}$, то и их пересечение $\cap_{i=1}^{\infty} A_i$ также принадлежит $\mathcal{A}$.

Поэтому *сигма-алгебру* можно также определить как такое семейство подмножеств, которое замкнуто относительно объединения и пересечения счетного числа подмножеств, а также перехода от множества к его дополнению.

Подмножества из $\mathcal{A}$ называются *событиями*, а $\mathbb{P}(A)$ есть вероятность того, что произойдет событие $A \in \mathcal{A}$, точнее произойдет такое элементарное событие $\omega \in \Omega$, что $\omega \in A$. Поэтому вероятностную меру $\mathbb{P}$ также называют *распределением вероятностей* на множестве Ω. Множества не из $\mathcal{A}$ не являются событиями.

Понятно, что нам хотелось бы определить вероятностное пространство с как можно большим числом событий. Поэтому очевидным кандидатом на роль «наилучшей» сигма-алгебры $\mathcal{A}$ является множество 2^{Ω} всех подмножеств множества Ω. Но для несчетных подмножеств (таких, как числовая прямая $\mathbb{R}$ или отрезок $[0,1]$) на такой большой сигма-алгебре незозможно определить вероятностную меру $\mathbb{P}$.

Обозначим через $\Delta(\Omega)$ семейство всех вероятностных мер на множестве Ω. Любая вероятностная мера $\mathbb{P} \in \Delta(\Omega)$ определяет вероятностное пространство $(\Omega, \mathcal{A}, \mathbb{P})$, где $\mathcal{A} \subseteq 2^{\Omega}$ есть область определения функции $\mathbb{P}$. Подмножества $A \subseteq \Omega$, на которых определена мера $\mathbb{P}$ ($A \in \mathcal{A}$), называются *измеримыми множествами*.

Если множество Ω счетное или конечное, то в качестве сигма-алгебры выбирают множество всех подмножеств множества Ω, т. е. $\mathcal{A} = 2^{\Omega}$. Обозначим через p_ω вероятность $\mathbb{P}(\{\omega\})$ наступления элементарного события $\omega \in \Omega$. Тогда из условия (i) получаем, что $\mathbb{P}(A) = \sum_{\omega \in A} p_\omega$ для всех $A \subseteq \Omega$. Такое вероятностное пространство называется *дискретным* и сокращенно задается парой $(\Omega, \{p_\omega\}_{\omega \in \Omega})$, где все вероятности p_ω неотрицательны и $\sum_{\omega \in \Omega} p_\omega = 1$.

Событие $A \cap B$ принято обозначать через $A \cdot B$ и называть произведением событий A и B. События A и B называются независимыми, если

$\mathbb{P}(A \cdot B) = \mathbb{P}(A)\,\mathbb{P}(B)$.

Вероятность наступления события A при условии, что произошло событие B обозначают через $\mathbb{P}(A|B)$ и называют *условной вероятностью события A относительно события B*. Имеет место *формула Байеса*:

$$\mathbb{P}(A \cdot B) = \mathbb{P}(B)\,\mathbb{P}(A|B).$$

Если A и B — независимые события, то $\mathcal{P}(A|B) = \mathcal{P}(A)$.

В.2. Случайные величины

Рассмотрим вероятностное пространство $(\Omega, \mathcal{A}, \mathbb{P})$. Функция $\xi : \Omega \to \mathbb{R}$ называется *случайной величиной*, если

$$\{\omega \in \Omega : \xi(\omega) \le x\} \in \mathcal{A} \quad \text{для любого } x \in \mathbb{R}. \tag{В.1}$$

Следовательно, мы можем вычислить вероятность

$$F(x) \stackrel{\text{def}}{=} \mathbb{P}(\xi \le x) \stackrel{\text{def}}{=} \mathbb{P}(\{\omega \in \Omega : \xi(\omega) \le x\}) \tag{В.2}$$

того, что случайная величина ξ примет значение, не превосходящее x. Определенная по формуле (В.2) функция $F : \mathbb{R} \to [0, 1]$ называется *функцией распределения* случайной величины ξ.

Поскольку сигма-алгебра $\mathcal{A}$ замкнута относсительно разности, то для $a < b$ множество

$$\{\omega \in \Omega : a < \xi(\omega) \le b\} = \{\omega \in \Omega : \xi(\omega) \le b\} \setminus \{\omega \in \Omega : \xi(\omega) \le a\}$$

также принадлежит $\mathcal{A}$, и поэтому мы можем вычислить вероятность того, что значение случайной величины ξ принадлежит отрезку $(a, b]$:

$$\mathbb{P}(a < \xi \le b) \stackrel{\text{def}}{=} \mathbb{P}(\{\omega \in \Omega : a < \xi(\omega) \le b\}) = F(b) - F(a). \tag{В.3}$$

Итак, случайная величина определена как функция, заданная на выборочном пространстве вероятностного пространства. Но, за исключением, пожалуй, только некоторых дискретных вероятностных пространств, чаще всего мы мало что знаем об этом вероятностном пространстве. Например, изменение цены акции нефтяной компании зависит от множества факторов, о некоторых их которых мы не имеем ни малейшего представления: будущая цена акций компании, добывающей нефть в Африке, зависит от поведения неизвестных нам террористических групп, которые могут взорвать нефтепровод. Следовательно, случайная величина — это специфическая функция, область определения которой может быть неизвестной. На практике, чаще всего мы знаем только функцию распределения случайной величины, которая позволяет судить о *частоте*, с которой данная функция принимает значения из любого интересующего нас интервала.

B.2.1. Математическое ожидание, дисперсия и стандартное отклонение

Равенство (B.3) позволяет определить *математическое ожидание* ограниченной случайной величины $\xi : \Omega \to \mathbb{R}$ (существует константа $D \in \mathbb{R}$, что $\xi(\omega) < D$ для всех $\omega \in \Omega$) по формуле

$$E(\xi) \stackrel{\text{def}}{=} \int_{\Omega} \xi(\omega)\mathbb{P}(d\omega) \stackrel{\text{def}}{=} \lim_{\delta\to 0_+} \sum_{k=-\infty}^{\infty} k\delta\, \mathbb{P}(\{k\delta < \xi \le (k+1)\delta\}). \qquad (B.4)$$

Существование предела в формуле (B.4) обосновывается в рамкам *теории интегрирования Лебега*,[14] знание которой читателем этой книги не предполагается. Отметим также, что в данном учебном пособии рассматриваются только такие случайные величины, вычисление математического ожидания которых сводится к вычислению интеграла Римана, изучаемого в математическом анализе, или к вычислению суммы конечного или бесконечного числа слагаемых.

Для дискретного вероятностного пространства, когда множество Ω счетное или конечное, условие (B.1) выполняется для любой функции $\xi : \Omega \to \mathbb{R}$. Это значит, что любая функция $\xi : \Omega \to \mathbb{R}$ является (*дискретной*) случайной величиной и ее *математическое ожидание* вычисляется по формуле:

$$E(\xi) = \sum_{\omega\in\Omega} \xi(\omega)\, p_\omega,$$

при условии, что ряд в правой части формулы сходится абсолютно, т. е.

$$\sum_{\omega\in\Omega} |\xi(\omega)|\, p_\omega < \infty.$$

[14] Анри Лебег (1875–1941) — основоположник современной теории интегрирования.

Если рассматриваемый ряд не сходится абсолютно, то говорят, что математического ожидания не существует[15]. Понятно, что математическое ожидание всегда существует, если множество Ω конечное.

Если функция распределения $F(x)$ случайной величины ξ является непрерывно дифференцируемой[16], то говорят, что случайная величина имеет *плотность* $f(x) = F'(x)$ и тогда

$$F(x) = \int_{-\infty}^{x} f(u)\,du, \tag{B.5}$$

$$\mathbb{P}(a \le \xi \le b) = \int_{a}^{b} f(u)\,du, \tag{B.6}$$

$$\int_{-\infty}^{\infty} f(u)\,du = 1, \tag{B.7}$$

$$E(\xi) = \int_{-\infty}^{\infty} uf(u)\,du. \tag{B.8}$$

Говорят, что *математическое ожидание* $E(\xi)$ существует, если интеграл в определении $E(\xi)$ сходится абсолютно, т. е.

$$\int_{-\infty}^{\infty} |u|f(u)\,du < \infty.$$

Если для функции $\phi : \mathbb{R} \to \mathbb{R}$ функция $\phi(\xi(\omega))$ также является случайной величиной, то

$$E(\phi(\xi)) = \int_{-\infty}^{\infty} \phi(u)f(u)\,du \tag{B.9}$$

при условии, что интеграл сходится абсолютно.

B.2.2. Совместное распределение случайных величин

Функция $F : \mathbb{R}^n \to \mathbb{R}$ *совместного распределения* случайных величин $\xi_1, \ldots, \xi_n : \Omega \to \mathbb{R}$ определяется следующим образом:

$$F(x) = F(x_1, \ldots, x_n) \stackrel{\text{def}}{=} \mathbb{P}(\{\omega \in \Omega : \xi_1(\omega) \le x_1, \ldots, \xi_n(\omega) \le x_n\})$$

[15] Требование абсолютной сходимости связано с тем, что математическое ожидание не должно зависеть от порядка суммирования. По теореме Римана можно переставить слагаемые неабсолютно сходящегося ряда таким образом, чтобы преобразованный ряд имел своей суммой любое заданное число, конечное или равное $\pm\infty$.

[16] На самом деле нам достаточно предположить, что $F(x)$ непрерывно дифференцируема почти всюду (например, за исключением конечного числа точек).

Случайные величины $\xi_1, \ldots, \xi_n$ называются *независимыми*, если

$$F(x) = F_1(x_1) \times \cdots \times F_n(x_n) \quad \text{для всех } x \in \mathbb{R}^n,$$

где F_j — функция распределения случайной величины x_j, $j = 1, \ldots, n$.

B.3. H

Приложение C
Графы

Графом называется пара $G = (V, E)$, где V — конечное множество, элементы которого называются *вершинами*, а E — это множество *ребер*, каждое из которых представляется парой (v, w) вершин из V. Порядок следования вершин не имеет значения: пары (v, w) и (w, v) задают одно и то же ребро. Если $e = (v, w) \in E$, то говорят, что вершины v и w *смежны*, и что ребро e *инцидентно* вершинам v и w. *Степенью* вершины v, обозначается $\deg(v)$, в графе G называется количество инцидентных ей ребер. В качестве упражнения докажите, что сумма степеней всех вершин равна удвоенному числу ребер: $\sum_{v \in V} \deg(v) = 2 \cdot |E|$[17].

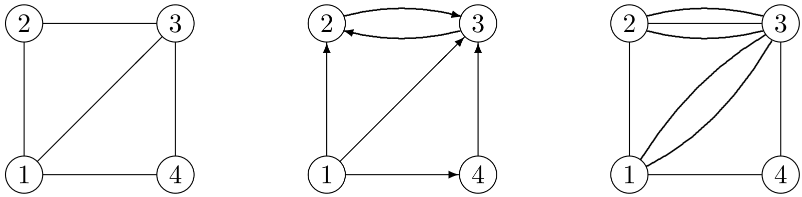
Графы небольшого размера удобно представлять рисунком на плоскости. Например, граф $G = (V, E)$ с множеством вершин $V = \{1, 2, 3, 4\}$ и множеством ребер $E = \{(1, 2), (1, 3), (1, 4), (2, 3), (3, 4)\}$ изображен на рис. C.1, а. Здесь степень вершин 1 и 3 равна 3, а степень вершин 2 и 4 равна 2.

Ориентированным графом (*орграфом*) называется пара $G = (V, E)$, где V — конечное множество вершин, а E — это множество упорядоченных пар вершин. Теперь элементы $e = (v, w)$ множества E называются *дугами*. Также говорят, что дуга $e = (v, w)$ выходит из вершины v и входит в вершину w. *Степенью исхода* вершины v, обозначается $\mathrm{outdeg}(v)$, называется количество дуг, выходящих из v. *Степенью захода* вершины v, обозначается $\mathrm{indeg}(v)$, называется количество дуг, входящих в v. На рис. C.1, б изображен орграф $G = (V, E)$ с множеством вершин $V = \{1, 2, 3, 4\}$ и множеством дуг $E = \{(1, 2), (1, 3), (1, 4), (2, 3), (3, 2), (4, 1)\}$.

Иногда полезно рассматривать *мультиграфы*, т. е. графы (орграфы) с кратными (или параллельными) ребрами (дугами). Пример мульти-

[17] Это равенство известно как *лемма о рукопожатиях* из за следующей задачи. На приеме каждый гость подсчитывает, сколько кукопожатий он сделал. По окончании приема вычисляется сумма рукопожатий каждого из гостей. Нужно доказать, что полученная сумма *четна*.

графа изображен на рис. C.1, в.



а — граф б — орграф в — мультиграф

Рис. C.1. Примеры графов

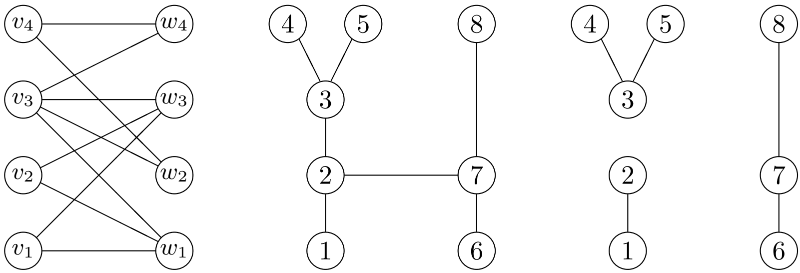
Последовательность вершин $P = (s = v_0, v_1, \ldots, v_k = t)$ называется *путем* из вершины s в вершину t *длины* k в графе (орграфе) $G = (V, E)$, если $(v_{i-1}, v_i) \in E$ для $i = 1, \ldots, k$. Путь называется *простым*, если в нем нет повторяющихся вершин. Замкнутый (когда $s = t$) путь называют *циклом*. *Простой цикл* не имеет повторяющихся вершин.

C.1. Специальные типы графов

Граф $G = (V, E)$ называется *двудольным*, если его множество вершин V можно разбить на два подмножества (доли) V_1 и V_2 ($V_1 \cap V_2 = \emptyset$, $V_1 \cup V_2 = V$) так, что каждое ребро из E инцидентно одной вершине из V_1 и одной вершине из V_2, т. е., если $(v, w) \in E$, то $v \in V_1$, а $w \in V_2$. Пример двудольного графа изображен на рис. C.2, а. Нетрудно доказать, что граф является двудольным тогда и только тогда, когда он не содержит циклов нечетной длины.

Граф (орграф) G называется *полным*, если любая (упорядоченная) пара его вершин соединена ребром (дугой). Полный граф с $V = \{1, \ldots, n\}$ принято обозначать через K_n. Двудольный граф $G = (V_1 \cup V_2, E)$ называется *полным*, если любая вершина из V_1 соединена ребром с любой вершиной из V_2. Полный двудольный граф с $V_1 = \{1, \ldots, m\}$ и $V_2 = \{1, \ldots, n\}$ принято обозначать через $K_{m,n}$.

Еще одним известным классом графов являются *деревья*. Граф называется *связным*, если между любыми его двумя вершинами имеется путь. *Дерево* — это связный граф без циклов. Пример дерева изображен на рис. C.2, б. *Лес* — это граф без циклов (или ациклический граф).



а — двудольный граф б — дерево в — лес (из 3-х деревьев)

Рис. C.2. Примеры специальных графов

Можно также сказать, что лес — это множество вершинно не пересекающихся деревьев. Пример леса изображен на рис. C.2, в.

Теорема C.1. *Для графа $G = (V, E)$ следующие условия эквивалентны:*

1) *G является деревом;*
2) *G — связный граф с $|V| - 1$ ребрами;*
3) *G не содержит циклов, но при добавлении любого нового ребра к G в нем появится единственный цикл.*

Орграф $G = (V, E)$ называется *ориентированным деревом* (или *ордеревом*), если $|E| = |V| - 1$ и в каждую вершину входит не более одной дуги. Единственная вершина в ордереве, в которую не входят дуги, называется *корнем*. Вершины, из которых не выходят дуги, называются *листьями*.

Покрывающим (или *остовным*) *деревом* (соотв., *ордеревом*) графа (соотв., орграфа) $G = (V, E)$ называется такой его подграф $T = (V, E')$, который является деревом (соотв., ордеревом).

C.2. Примеры самых известных задач теории графов

Многие задачи теории графов — это результат представления в графовых терминах практических задач, головоломок, игр и т. д. В этом разделе представлены несколько самых известных графых задач. Ряд других графовых задач, которые имеют отношение к сетевой оптимизации, рассматриваются в главе 6.

С.2.1. Эйлеровы графы

На реке Преголь в Кенигсберге было два острова, которые соединялись между собой и с берегами реки семью мостами, как показано на рис. С.3, а. Задача заключалась в том, чтобы, начав двигаться с одного из участков суши, помеченных на рисунке буквами A, B, C и D, пройти по каждому мосту ровно один раз и в результате вернуться в исходную точку.

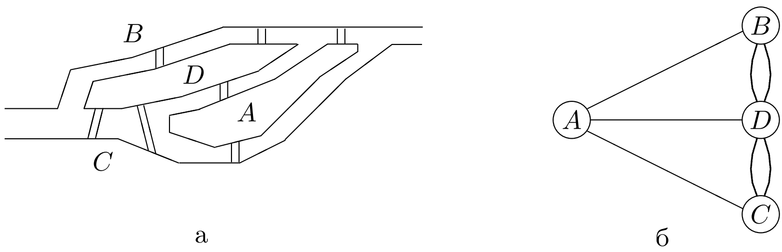


Рис. С.3. Задача о кенигсбергских мостах

Многие (способные логически рассуждать) были убеждены, что такого маршрута не существует. Но лишь в 1736 г. великий математик Эйлер строго доказал это предположение, представив схему мостов мультиграфом, изображенным на рис. С.3, б. В этом мультиграфе нужно найти цикл (не обязательно простой), который проходит по каждому ребру ровно один раз. В последствии такие циклы стали называть *эйлеровыми циклами*, а графы (мультиграфы), содержащие эйлеровы циклы, — *эйлеровыми графами* (*мультиграфами*).

Теорема С.2 (Эйлера). *Мультиграф G эйлеров тогда и только тогда, когда он связен и степень каждой его вершины четная.*

Поскольку степень вершины A в мультиграфе на рис. С.3, б нечетна (равна трем), то этот мультиграф не имеет эйлерового цикла, и, следовательно, задача о кенигсбергских мостах также не имеет решения.

C.2.2. Задача коммивояжера

Рассмотрим граф $G = (V, E)$. Цикл, содержащий все $n = |V|$ вершин графа, называется *гамильтоновым.* Граф G называется *гамильтоновым*, если он содержит гамильтонов цикл. Проверка того, что заданный граф является гамильтоновым, является одной из самых знаменитых задач теории графов.

В 1859 г. известный математик У. Гамильтон головоломку, в которой требовалось найти обход всех вершин додекадрона, посещая каждую вершину не более одного раза. Эта задача эквивалентна задаче поиска гамильтонова цикла в графе, представленного на рис. C.4.

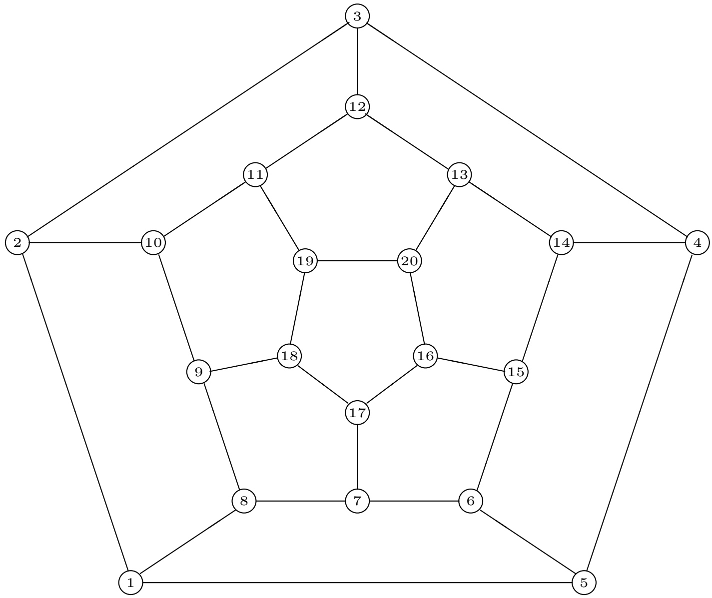


Рис. C.4. Граф головоломки Гамильтона

В качестве иллюстрации рассмотрим более сложную ситуацию. При дворе короля Артура проживали 500 рыцарей, не все из которых ладили между собой. Из-за этого во время обеда, где все рыцари сидели за круглым столом, часто возникали драки между сидящими рядом рыцарями. Королю Артуру это не нравилось, и он приказал своему магу Мерлину рассадить рыцарей таким образом, чтобы рядом не сидели враждующие рыцари. Задача Мерлина формулируется как задача поиска гамильтонова цикла в графе, в котором вершины представляют рыцарей, и две вершины соединены ребром, если соответствующие им рыцари не враждуют между собой.

В более общей *задаче о минимальном гамильтоновом цикле* каждому ребру $(v, w) \in E$ приписана стоимость (длина) $c(v, w)$, и нам нужно найти гамильтоном цикл $\Gamma = (v_0, v_1, \dots, v_n = v_0)$ минимальной стоимости $c(\Gamma) \stackrel{\text{def}}{=} \sum_{i=1}^{n} c(v_{i-1}, v_i)$.

Задача коммивояжера — это задача о минимальном гамильтоновом цикле в полном графе. Задача получила свое название из-за следующей интерпретации. Вершины графа представляют некоторые города, а стоимости $c(v, w)$ — это расстояния между городами. Коммивояжер, начиная из города, в котором он проживает, хочет посетить каждый из остальных $n - 1$ городов ровно один раз и вернуться обратно в родной город, при этом длина его маршрута должна быть минимальной.

C.2.3. Задача о максимальной клике

Граф $H = (\bar{V}, \bar{E})$ называется *подграфом* графа $G = (V, E)$, если $\bar{V} \subseteq V$ и $\bar{E} \subseteq E$. Максимальный (по включению) полный подграф графа G называется *кликой*. На практике часто встречается *задача о максимальной клике*, целью в которой является поиск клики с максимальным количеством вершин. Для примера, пусть вершины графа представляют некоторую группу людей, и две вершины соединены ребром, если соответствующие им люди знакомы друг с другом. Мы решаем задачу о максимальной клике, когда ходим найти наибольшую подгруппу людей попарно знакомых друг с другом

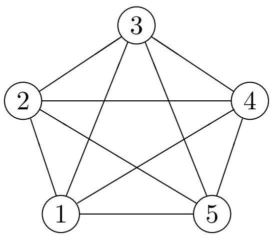
C.2.4. Раскраска графа и проблема четырех красок

В какое минимальное число цветов можно раскрасить вершины заданного графа, чтобы никакие две смежные вершины не были окрашены в один цвет. Так формулируется *задача о раскраске графа*. Самый знаменитый частный случай данной задачи, известный как *проблема четырех красок*, состоит в том, чтобы определить минимальное число цветов, необходимых для раскраски политической карты так, чтобы никакие две страны, имеющие общую границу, не были раскрашены в один цвет. Если представить каждую страну отдельной вершиной графа и соединить две вершины ребром, если соответствующие им страны имеют общую границу, то задача о раскраске карты представляется как задача о раскраске полученного графа.

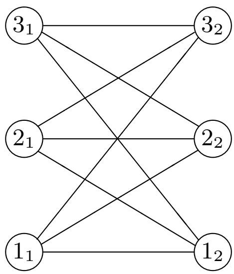
Нетрудно привести пример карты, для раскраски которой требуется четыре цвета. Долгое время гипотеза о том, что четырех цветов достаточно для раскраски любой карты оставалась недоказанной. Это было сделано Аппелем и Хакеном в 1976 г.[18] оригинальным способом: сначала доказательство гипотезы было сведено к рассмотрению достаточно большого числа частных случаев задачи, а затем была написана компьютерная программа, которая выполнила «раскраску» карт для каждого из выделенных случаев.

C.2.5. Укладка графа на плоскости

Как мы уже видели, графы можно рисовать на плоскости, причем, это можно сделать разными способами. Считается, что рисунок графа более привлекателен, если на нем количество пересечений ребер минимально. В идеале, хотелось бы полностью избежать пересечений ребер, но это не всегда возможно. Два самых «маленьких» графа, которые нельзя нарисовать на плоскости без пересечений ребер, — это графы K_5 и $K_{3,3}$, представленные на рис. C.5.



граф K_5



граф $K_{3,3}$

Рис. C.5. Примеры непланарных графов

Граф, который можно нарисовать на плоскости без пересечения ребер, называется *планарным.* Если граф G непланарен, то также непла-

[18] K.I. Appel, W. Haken. Every planar map is four-colorable. *Bull. Am. Math. Soc.* **82** (1976) 711–712.

нарен и граф G', который получается из исходного переименованием вершин и заменой нескольких его ребер простыми путями. Графы G и G' называются *гомеоморфными*.

Теорема C.3 (Куратовского). *Граф G планарен тогда и только тогда, когда он не содержит подграфов, гомеоморфных K_5 и $K_{3,3}$.*

Приложение D
Сложность вычислений

D.1. Сложность алгоритмов

Сложность алгоритма измеряется функцией от *размера задачи*, т. е. количества бит в памяти компьютера, неабходимых для представления исходных данных решаемой задачи.

Размером рационального числа $\alpha = \frac{p}{q}$ (p и q — взаbмно простые целые числа), *размером* рационального вектора $b \in \mathbb{Q}^n$, *размером* рациональной $m \times n$-матрицы $A = [a_{ij}]$ называются величины

$$\begin{aligned}
\text{size}(\alpha) &\stackrel{\text{def}}{=} 1 + \lceil \log(|p| + 1) \rceil + \lceil \log(|q| + 1) \rceil, \\
\text{size}(b) &\stackrel{\text{def}}{=} n + \sum_{i=1}^{n} \text{size}(b_i), \\
\text{size}(A) &\stackrel{\text{def}}{=} mn + \sum_{i=1}^{m} \sum_{j=1}^{n} \text{size}(a_{ij}),
\end{aligned} \tag{D.1}$$

В качестве меры сложности алгоритма чаще всего рассматривают две характеристики: *время его работы* и *объем используемой памяти*, которые выражаются как функции от размера задачи. Так как время работы алгоритма, обычно, не меньше объема памяти, то в дальнейшем под *сложностью алгоритма*, главным образом, мы будем понимать его *временную сложность* (время работы).

Вычислительные алгоритмы, которые мы будем изучать, решают некоторую задачу, выполняя определенную последовательность *элементарных арифметических и логических операций* (сложений, вычитаний, произведений, делений и сравнений). *Временная сложность* таких алгоритмов определяется как количество элементарных операций, которые он выполняет. Такой подход к оценке сложности алгоритмов называется *алгебраическим*. Алгебраический подход игнорирует дискретность данных в памяти компьютера (там нет действительных чисел, а только рациональные). В памяти компьютера под запис числа отводится фиксированное количество битов. Это ограничивает размеры чисел, над которыми арифметические операции выполняются точно (без округлений) и с одинакой скоростью. Если же размеры чисел, над которыми выполняются арифметические операции, или результаты этих операций превосходят размеры компьютерной разрядной сетки, то для точного выполения арифметических операций нужно использовать библиотеку алгоритмов для выполнения арифметический операций с длинными (большими) числами.

Так, существуют алгоритмы выполнения всех арифметических операций битовой сложности $\mathrm{const}\, l \log l \log(\log l)$, где l — максимальный размер числа, участвующего в арифметической операции. Чтобы получить более точную *битовую* оценку сложности алгоритма, сначала нужно оценить максимальный размер l чисел, над которыми алгоритм выполняет арифметические операции, а затем алгебраическую сложность алгоритма умножить на битовую сложность арифметический операций над числами размера l. И все же, за исключением достаточно редких случаев, когда в вычислениях задествованы большие числа, или когда нельзя проводить вычисления с огруглением результатов, в качестве сложности алгоритма рассматривают его алгебраическую сложность.

Анализ сложности алгоритма обычно проводят по наиболее сложному примеру решаемой задачи. Это означает, что в качестве сложности алгоритма принимается его максимальная сложность на примерах задач одинакового размера. Мы анализируем сложность алгоритма, главным образом, чтобы понять может ли он решать задачи большого размера. Поэтому при оценке сложности алгоритмов ограничиваются проведением так называемого *асимптотического анализа*), в которм игнорируются константные множители. Это позволяет не только упростить анализ, но и делает его независимым от деталей представления исходных данных (*входа*) задачи.

D.2. Полиномиальные алгоритмы

Строгие формализации понятия алгоритм, подобные на машину Тьюринга, прывели математиков 30-х годов 20-го века до деления всех задач на *алгоритмически разрешимые* (для которых существуе алгоритм) и *неразрешимые* (для которых нет алгоритмов решения)[19]. Современные компьютеры поставили перед математиками другие задачи. Главная из них — это разработка эффективных алгоритмов решения разных задач. Под*эффективными алгоритмами* обычно понимают полинамиальные алгоритмы.

Говорят, что алгоритм является *полиномиальным*, если время его работы ограничено некоторым полиномом от размера задачи. Также говорят, что задача *полиномиально разрешима*, если для ее решения существует полиномиальный алгоритм. Так как все элементарные арифметические операции можно выполнить за полиномиальное время, то для доказательства полиномиальности некоторого алгоритма достаточно показать, что он выполняет полиномиальное от размера L задачи количество операций над числами, размер которых ограничен полиномом от L.

В качестве примера рассмотрим известную всем задачу *факторизации* натурального числа n, в которой нужно найти делитель n, если такой существует. Так как вход этой задачи задается только одним числом n, то яе размер равен $\mathrm{size}(n) = O(\log n)$. Простейший алгоритм, который по очереди делит n на $2, \dots, \lfloor \frac{n}{2} \rfloor$, выполняет $O(n) = O(2^{\log n})$ арифметических операции и поэтому не является полиномиальным. На данный момент вопрос о полиномиальной разрешимости задачи факторизации остается открытым. Положительный ответ на этот вопрос может иметь серьезные последствия для современной криптографии, поскольку надежность некоторых современных криптосистем с публичным ключем основана на предположении, что задача факторизации не может быть решена за полиномиальное время.

---

[19] Типичным примером неразрешимой задачи является *проблема остановки*: для конкрнтной програмы и исходных данных нужно определить, завршит ли работу эта программа или нет.

Литература

1. Вагнер Г. Основы исследования операций (в 3-х томах). — М.: Мир, 1972-1973.

2. Вентцель Е. С. Исследование операций: задачи, принципы, методология. — М.: Наука, 1988.

3. Интрилигатор М. Математические методы оптимизации и экономическая теория. — М.: Айрис Пресс, 2002.

4. Исследование операций. Т. 1. Методологические основы и математические методы. Т. 2. Модели и применения. Под ред. Дж. Моудера, С. Элмаграби. — М.: Мир, 1981.

5. Карманов В. Г. Математическое программирование. — М.: Наука, 1975.

6. Костевич Л. С. Математическое программирование: Информационные технологии оптимальных решений. — Мн. : ООО "Новое знание 2003.

7. Мину М. Математическое программирование. — М.: Наука, 1990.

8. Писарук Н. Н. Модели и методы смешанно-целочисленного программирования. — Мн.: Изд-во БГУ, 2010.

9. Сакович В. А. Исследование операций. — Мн.: Вышэйшая школа, 1985.

10. Таха Х. Введение в исследование операций. В 2-х кн. — М.: Мир, 1985.

11. Филлипс Д., А. Гарсиа-Диас. Методы анализа сетей. — М.: Мир, 1984.

12. Чжун К.Л., Ф. АитСахлиа. Элементарный курс теории вероятностей. Стохастические процессы и финансовая математика. — М.: Бином, 2007.

13. Birge J. R., F. V. Louveaux. Introduction to stochastic programming. Springer Verlag, New York, 1997.

Предметный указатель